

A modal typing system for self-referential programs and specifications¹

Hiroshi Nakano

*Department of Applied Mathematics and Informatics,
Ryukoku University, Seta, Otsu, Shiga 520-2194, Japan
Email: nakano@math.ryukoku.ac.jp*

Abstract

This paper proposes a modal typing system that enables us to handle self-referential formulae, including ones with negative self-references, which on one hand, would introduce a logical contradiction, namely Russell's paradox, in the conventional setting, while on the other hand, are necessary to capture a certain class of programs such as fixed-point combinators and objects with so-called binary methods in object-oriented programming. The proposed system provides a basis for axiomatic semantics of such a wider range of programs and a new framework for natural construction of recursive programs in the proofs-as-programs paradigm.

Keywords: typed lambda calculi, self-reference, proofs-as-programs, modal logic, fixed-point combinators, object oriented programming, termination verification.

1. Introduction

Although recursion, or self-reference, plays an indispensable role in both programs and their specifications, it also introduces serious difficulties into their formal treatment. It is still far from obvious how to capture it in an axiomatic semantics such as the formulae-as-types notion of construction [17]. Only a rather restricted class of recursive programs (and specifications) has been captured in this direction as (co)inductive proofs over the (co)inductive data structures (see e.g., [8, 14, 23, 18, 28]), and, for example, negative self-references, which would be necessary to handle a certain range of programs such as fixed-point combinators and objects with so-called binary methods in object-oriented programming, still remain out of the scope. In this paper, a modal logic that provides a basis for capturing such a wider range of programs in the proofs-as-programs paradigm is proposed. The logic is presented as a modal typing system with recursive types for the purpose of presentation. Its soundness with respect to a realizability interpretation, which implies the convergence of well-typed programs, is shown.

Difficulty in binary-methods

Consider, for example, the specification $\mathbf{Nat}(n)$ of objects that represent a natural number n with a *method* which returns an object of $\mathbf{Nat}(n+m)$ when one of $\mathbf{Nat}(m)$ is given. It could be represented by a self-referential specification such as

$$\mathbf{Nat}(n) \equiv ((n = 0) + (n > 0 \wedge \mathbf{Nat}(n-1)) \times (\forall m. \mathbf{Nat}(m) \rightarrow \mathbf{Nat}(n+m))),$$

where we assume that n and m range over the set of natural numbers; $+$, \times and \rightarrow are type constructors for direct sums, direct products and function spaces, respectively; \wedge and \forall have standard logical (annotative) meanings. Although it is not obvious whether this self-referential specification is meaningful in a certain

¹Research supported in part by 2014 Special Researcher Program of Ryukoku University. Some preliminary results to this paper have been presented in [21, 22].

mathematical sense, it could be a first approximation of the specification we want since this can be regarded as a refined version of recursive types which have been widely adopted as a basis for object-oriented type systems [1, 5]. At any rate, if we define an object **0** as

$$\mathbf{0} \equiv \langle \mathbf{i}_1 *, \lambda x. x \rangle,$$

then it would satisfy $\mathbf{Nat}(0)$, where \mathbf{i}_1 is the injection into the first summand of direct sums and $*$ is a constant. We assume that any program satisfies annotative formulae such as $n = 0$ whenever they are true. We can easily define a function that satisfies $\forall n. \forall m. \mathbf{Nat}(n) \rightarrow \mathbf{Nat}(m) \rightarrow \mathbf{Nat}(n+m)$ as

$$\mathbf{add} \ x \ y \equiv \mathbf{p}_2 \ x \ y,$$

or

$$\mathbf{add}' \ x \ y \equiv \mathbf{p}_2 \ y \ x,$$

where \mathbf{p}_2 extracts the second components, i.e., the method of addition in this particular case, from pairs. We could also define the successor function as a recursive program as

$$\mathbf{s} \ x \equiv \langle \mathbf{i}_2 \ x, \lambda y. \mathbf{add} \ x \ (\mathbf{s} \ y) \rangle$$

or

$$\mathbf{s}' \ x \equiv \langle \mathbf{i}_2 \ x, \lambda y. \mathbf{add}' \ x \ (\mathbf{s}' \ y) \rangle.$$

In spite of the apparent symmetry between \mathbf{add} and \mathbf{add}' , which are both supposed to satisfy the same specification, the computational behaviors of \mathbf{s} and \mathbf{s}' are completely different. We can observe that \mathbf{s} works as expected, but \mathbf{s}' does not.

For example, $\mathbf{p}_2 \ (\mathbf{s} \ \mathbf{0}) \ \mathbf{0}$ would be evaluated as

$$\begin{aligned} \mathbf{p}_2 \ (\mathbf{s} \ \mathbf{0}) \ \mathbf{0} &\rightarrow (\lambda y. \mathbf{add} \ \mathbf{0} \ (\mathbf{s} \ y)) \ \mathbf{0} \\ &\rightarrow \mathbf{add} \ \mathbf{0} \ (\mathbf{s} \ \mathbf{0}) \\ &\rightarrow \mathbf{p}_2 \ \mathbf{0} \ (\mathbf{s} \ \mathbf{0}) \\ &\rightarrow (\lambda x. x) \ (\mathbf{s} \ \mathbf{0}) \\ &\rightarrow \mathbf{s} \ \mathbf{0}, \end{aligned}$$

whereas

$$\begin{aligned} \mathbf{p}_2 \ (\mathbf{s}' \ \mathbf{0}) \ \mathbf{0} &\rightarrow (\lambda y. \mathbf{add}' \ \mathbf{0} \ (\mathbf{s}' \ y)) \ \mathbf{0} \\ &\rightarrow \mathbf{add}' \ \mathbf{0} \ (\mathbf{s}' \ \mathbf{0}) \\ &\rightarrow \mathbf{p}_2 \ (\mathbf{s}' \ \mathbf{0}) \ \mathbf{0} \\ &\rightarrow \dots, \end{aligned}$$

and more generally, for any objects x and y of $\mathbf{Nat}(n)$ (for some n),

$$\mathbf{p}_2 \ (\mathbf{s}' \ x) \ y \rightarrow \dots \rightarrow \mathbf{p}_2 \ (\mathbf{s}' \ y) \ x \rightarrow \dots \rightarrow \mathbf{p}_2 \ (\mathbf{s}' \ x) \ y \rightarrow \dots$$

This sort of divergence would also be quite common in (careless) recursive definitions of programs even if we did not have to handle object-oriented specifications like $\mathbf{Nat}(n)$. The peculiarity here is the fact that the divergence is caused by a program, \mathbf{add}' , which is supposed to satisfy the same specification as \mathbf{add} . This example shows such a loss of the compositionality of programs with respect to the specifications that imply their termination, or convergence. It also suggests that, to overcome this difficulty, \mathbf{add} and \mathbf{add}' should have different specifications, and accordingly the definition of $\mathbf{Nat}(n)$ should be revised in some way in order to force it.

$\lambda\mu$ and its logical inconsistency

The typing system $\lambda\mu$ (cf. [3]) is a simply-typed lambda calculus with recursive types, where any form of self-references, including negative ones, is permitted. A non-trivial model for such unrestricted recursive types was developed by MacQueen, Plotkin and Sethi [20], and has been widely adopted as a theoretical basis for object-oriented type systems [1, 5].

On the other hand, it is well known that logical formulae with such unrestricted self-references would introduce a contradiction (variant of Russell's paradox). Therefore, logical systems must have certain restrictions on the forms of self-references (if ever allowed) in order to keep themselves sound; for example, μ -calculus [24, 19] does not allow negative self-references (see also [13]).

Through the formulae-as-types notion, this paradox corresponds to the fact that every type of $\lambda\mu$ is inhabited by a diverging program which does not produce any information; for example, the λ -term $(\lambda x. xx)(\lambda x. xx)$ can be typed with every type in $\lambda\mu$. Therefore, even with the model mentioned above, types can be regarded only as partial specifications of programs, and that is considered the reason why we lost the compositionality of programs in the $\mathbf{Nat}(n)$ case, where we regarded convergence of programs as a part of their specifications. This shows a contrast with the success of $\lambda\mu$ as a basis for type systems of object-oriented program languages, where the primary purpose of types, i.e., coarse specifications, is to prevent run-time type errors, and termination of programs is out of the scope.

The logical inconsistency of $\lambda\mu$ also implies that no matter how much types, or specifications, are refined, convergence of programs cannot be expressed by them, and must be handled by endowing the typing system with some facilities for discussing computational properties of programs. For example, Constable et al. adopted this approach in their pioneering works to incorporate recursive definitions and partial objects into constructive type theory [9, 10]. However, in this paper, we will pursue another approach such that types themselves can express convergence of programs.

Towards the approximation modality

Suppose that we have a recursive program f defined by

$$f \equiv F(f),$$

and want to show that f satisfies a certain specification S . Since the denotational meaning of f is given as the least fixed point of F , i.e., $f = \sup_{n < \omega} F^n(\perp)$, a possible way to do that would be to apply Scott's fixed-point induction [26] by showing the following:

- \perp satisfies S ,
- $F(x)$ satisfies S provided that x satisfies S , and
- S is chain closed.

However, this does not suffice for our purpose if S includes some requirement about the convergence of f , because obviously \perp , or even $F^n(\perp)$, could not satisfy the requirement. So we need more refined approach. The failure of the naive fixed-point induction above suggests that the specification to be satisfied by each $F^n(\perp)$ inherently depends on n , and the requirement concerning its convergence must become stronger when n increases. This leads us to a layered version of the fixed-point induction scheme as follows: in order to show that f satisfies S , it suffices to find an infinite sequence S_0, S_1, S_2, \dots of properties, or (virtual) specifications, such that

- (i) $S = \bigcap_{n < \omega} S_n$,
- (ii) $S_{n+1} \subseteq S_n$,
- (iii) \perp satisfies S_0 ,
- (iv) $F(x)$ satisfies S_{n+1} provided that x satisfies S_n , and
- (v) S_n is chain closed.

For, since $F^n(\perp) \in S_n$ for every n by (iii) and (iv), we get $F^k(\perp) \in S_n$ for every $k \geq n$ by (ii). This and (v) imply $f \in S_n$ for every n , and consequently $f \in S$ by (i).

In this scheme, the sequence S_0, S_1, S_2, \dots can be regarded as a successive approximation of S , and F a (higher-order) program which constructs a program that satisfies S_{n+1} from one that satisfies S_n . It should be also noted that F works independently of n . This uniformity of F over n leads us to consider a formalization of this scheme in a modal logic, where the set of possible worlds (in the sense of Kripke semantics) consists of all non-negative integers, and S_n in the induction scheme above corresponds to the interpretation of S in the world n . We now write $x \mathbf{r}_k S$ to denote the fact that x satisfies the interpretation of S in the world k , and define a modality, say \bullet , as

$$x \mathbf{r}_k \bullet S \quad \text{iff} \quad k = 0 \quad \text{or} \quad x \mathbf{r}_{k-1} S.$$

Condition (ii) of the induction scheme says that $x \mathbf{r}_k S$ implies $x \mathbf{r}_l S$ for every $l \leq k$; in other words, the interpretation of specifications should be *hereditary* with respect to the accessibility relation $>$. In such a modal framework, the specification to be satisfied by F can be represented by $\bullet S \rightarrow S$ provided that the \rightarrow -connective is interpreted in the standard way in each world, and our induction scheme can be rewritten as

$$\text{if } \perp \mathbf{r}_0 S \text{ and } F \mathbf{r}_k \bullet S \rightarrow S \text{ for every } k > 0, \text{ then } f \mathbf{r}_k S \text{ for every } k.$$

Furthermore, if we assume that S_0 is a trivial specification which is satisfiable by any program, then, shifting the possible worlds downwards by one, we can simplify this to

$$\text{if } F \mathbf{r}_k \bullet S \rightarrow S \text{ for every } k, \text{ then } f \mathbf{r}_k S \text{ for every } k. \quad (*)$$

Although this assumption about S_0 somewhat restricts our choice of the sequence S_0, S_1, S_2, \dots , it could be thought rather reasonable because S_0 must be an almost trivial specification that is even satisfiable by \perp . Note that S_{n+1} occurring in the induction now corresponds to the interpretation of S in the world n , and S_0 corresponds to the interpretation of $\bullet S$ in the world 0.

We so far considered the set of non-negative integers and the greater-than relation as the set of possible worlds and the accessibility relation, respectively. This setting can be easily generalized to any frame with a (conversely) well-founded accessibility relation. The standard interpretation of \rightarrow implies some fundamental properties concerning the \bullet -modality over such frames, which introduce a subsumption, or subtyping, relation over specifications into our modal framework. First, the hereditary interpretation of specifications implies the following property.

$$x \mathbf{r}_k S \quad \text{implies} \quad x \mathbf{r}_k \bullet S$$

Second, if we adopt the simple semantics of types, i.e., $x \mathbf{r}_k S \rightarrow \top$ for every x, S and k , where \top is the universe of (meanings of) programs, which is satisfiable by any program, then

$$x \mathbf{r}_k \bullet(S \rightarrow T) \quad \text{implies} \quad x \mathbf{r}_k \bullet S \rightarrow \bullet T.$$

Note that this is not always the case because we could consider non-extensional interpretations, e.g., F-semantics[15], in which $\lambda x. \perp \mathbf{r}_k S \rightarrow \top$ holds, but $\perp \mathbf{r}_k S \rightarrow \top$ does not.

Furthermore, if the accessibility relation is linear, i.e., not branching, such as the case of the greater-than relation between non-negative integers, there holds the following converse property.

$$x \mathbf{r}_k \bullet S \rightarrow \bullet T \quad \text{implies} \quad x \mathbf{r}_k \bullet(S \rightarrow T)$$

In such cases, the meaning of $\bullet S \rightarrow \bullet T$ and $\bullet(S \rightarrow T)$ are equivalent.

Specification-level self-references

This modal framework introduced for program-level self-references also provides a basis for specification-level self-references. Suppose that we have a self-referential specification such as

$$S = \phi(S).$$

As we saw in the **Nat**(n) case, negative reference to S in ϕ can introduce a contradiction in the conventional setting, and this is still true in our modal framework. However, in the world n , we can now refer to the interpretation of S in any world $k < n$ without worrying about the contradiction. That is, as long as S occurs only in scopes of the modal operator \bullet in ϕ , the interpretation of S is well-defined and given as a fixed point of ϕ , which is actually shown to be unique. For example, if S is defined as $S = \bullet S \rightarrow T$, then S could be interpreted in each world as follows.

$$\begin{aligned} S_0 &= \top \rightarrow T_0 \\ S_1 &= S_0 \cap ((\top \rightarrow T_0) \rightarrow T_1) \\ S_2 &= S_1 \cap ((S_0 \cap ((\top \rightarrow T_0) \rightarrow T_1)) \rightarrow T_2) \\ &\vdots \\ S_{n+1} &= S_n \cap (S_n \rightarrow T_{n+1}) \\ &\vdots \end{aligned}$$

where S_k and T_k are the interpretations of S and T in the world k , respectively, and the notations such as \top and \rightarrow are abused to denote their expected interpretations as well. This kind of self-references provides us a method to define the sequence S_0, S_1, S_2, \dots for the refined induction scheme when we derive properties of recursive programs, and the induction scheme would be useless if we did not have such a method.

In the following sections, we will see that this form of specification-level self-references is quite powerful, and captures a wide range of specifications including those which are not representable in the conventional setting such as ones for **add** and **add'** in the **Nat**(n) case. Furthermore, the modal version (*) of the induction scheme turns out to be derivable from other properties of the \bullet -modality and such self-referential specifications, where the derivation corresponds to fixed-point combinators, such as Curry's **Y**. This also gives us a way to construct recursive programs based on the proofs-as-programs notion.

The plan of the paper

The plan of the present paper is as follows. In the next section, the syntax of our type expressions, which can be considered as coarse specifications of programs, is given. The modal semantics of such type expressions is given in the manner of realizability in Section 3, in which we interpret types over Kripke frames with a (conversely) well-founded accessibility relation. We discuss formal derivability of equality and subsumption of types in Sections 4 and 5, respectively, and show their soundness with respect to the semantics. In Section 6, we introduce a typed λ -calculus $\lambda\mathbf{A}$ equipped with the modality and recursive types, and show its soundness and subject reduction property in Section 7. Then, in Section 8, we show convergence of well-typed λ -terms according to the forms of their type. In Section 9, we present some examples of program derivation by extending the pure typing system, where the **Nat**(n)-example presented in this section will be revisited. In Section 10, we consider $\lambda\mathbf{A}$ as a modal logic by ignoring left hand sides of “:” from typing, and show an interesting relationship to an intuitionistic version of the logic of provability.

2. Type expressions

We start with the syntax of type expressions. As a preparation for it, we first give one of *pseudo type expressions* **PTExp**, which are obtained by adding a unary type constructor \bullet to those of $\lambda\mu$, namely the simply typed λ -calculus extended with recursive types (cf. [3, 7]). Let **TVar** be a countably infinite set of type variable symbols X, Y, Z, \dots

Definition 2.1 (Pseudo type expressions). The syntax of **PTExp** is defined by

PTExp ::= TVar	(type variables)
PTExp \rightarrow PTExp	(function types)
\bullet PTExp	(approximative types)
μ TVar . PTExp	(recursive types).

We assume that \rightarrow associates to the right as usual, and each (pseudo) type constructor associates according to the following priority.

$$\text{(Low)} \quad \mu X. < \rightarrow < \bullet \quad \text{(High)}$$

For example, $\bullet\mu X.\bullet X \rightarrow Y \rightarrow Z$ is the same as $\bullet(\mu X.((\bullet X) \rightarrow (Y \rightarrow Z)))$. We use \top as an abbreviation for $\mu X.\bullet X$ and use $\bullet^n A$ to denote a (pseudo) type expression $\underbrace{\bullet \dots \bullet}_n A$, where $n \geq 0$. In the sequel, we use $A, B,$

C, D, \dots to denote (pseudo) type expressions, and denote the set of type variables occurring freely in A by $FTV(A)$ regarding a type variable X as bound in $\mu X.A$. We regard α -convertible (pseudo) type expressions as identical. We write $A[B_1/X_1, \dots, B_n/X_n]$ to denote the (pseudo) type expression obtained from A by substituting B_1, \dots, B_n for each free occurrence of X_1, \dots, X_n , respectively, with necessary α -conversion to avoid accidental capture of free type variables. We assume that $[B_1/X_1, \dots, B_n/X_n]$ associates with the preceding type expression with a higher priority than the modal operator \bullet .

As mentioned in Section 1, the modal type operator \bullet causes a shift of possible worlds, and makes a reference to the one-step coarser world, in which type expressions are interpreted as coarser specifications, i.e., larger sets of values or programs, than the ones in the present world. The (pseudo) type expression \top corresponds to the universe into which λ -terms are interpreted. Hence, every λ -term should have this type in $\lambda\mathbf{A}$. We adopt a so called *simple semantics* of types (cf. [15, 16]), by which the meaning of $A \rightarrow \top$ is identical to the one of \top . Thus, some syntactically different type expressions may have the same meaning as the one of \top . We call such (pseudo) type expressions \top -variants, which can be syntactically distinguished from others as follows.

Definition 2.2 (\top -variants). A (pseudo) type expression A is a \top -variant if and only if $A^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} X_i$ for some $n, m_0, m_1, m_2, \dots, m_n, X_1, X_2, \dots, X_n$ and i such that $1 \leq i \leq n$, $X_i \notin \{X_{i+1}, X_{i+2}, \dots, X_n\}$ and $m_i + m_{i+1} + m_{i+2} + \dots + m_n \geq 1$, where A^t is defined as follows.

$$X^t = X, \quad (A \rightarrow B)^t = B^t, \quad (\bullet A)^t = \bullet A^t, \quad (\mu X.A)^t = \mu X.A^t.$$

Note that A^t always has the form of $\bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} Y$ for some $n, m_0, m_1, m_2, \dots, m_n, X_1, X_2, \dots, X_n$ and Y , and it is decidable whether a pseudo type expression is a \top -variant or not. It will be shown that \top -variants are semantically identical to \top . We can also easily see that the following propositions hold.

- Proposition 2.3.**
1. X is not a \top -variant.
 2. $\bullet A$ is a \top -variant if and only if so is A .
 3. $A \rightarrow B$ is a \top -variant if and only if so is B .
 4. A is a \top -variant if and only if so is A^t .
 5. A is a \top -variant if and only if so is $A[Y/X]$.

Proof. Obvious from Definition 2.2. □

Unrestricted use of self-reference in type expressions causes logical contradictions, i.e., divergence of well-typed programs. So our typing system only allows self-references in coarser worlds, i.e., in scopes of the modal operator \bullet , or at positions where the references do not affect the meaning of the whole type expression. We say that such references are *proper*. The latter case is included so that the equivalence relations between type expressions (cf. Definitions 4.1 and 4.2) preserve properness. For example, the reference to X in $X \rightarrow \top$ is proper since $X \rightarrow \top$ is semantically identical to \top .

Definition 2.4 (Properness). A (pseudo) type expression A is *proper* in X if and only if X freely occurs only (a) in scopes of the \bullet -operator in A , or (b) in a \top -variant occurring in A . In other words,

1. A type variable Y is proper in X if and only if $Y \neq X$.
2. $\bullet A$ is proper in X .
3. $A \rightarrow B$ is proper in X if and only if (a) so are both A and B , or (b) B is a \top -variant.
4. Suppose that $X \neq Y$. Then, $\mu Y.A$ is proper in X if and only if (a) so is A or (b) $\mu Y.A$ is a \top -variant.

For example, $\bullet X$, $\bullet(X \rightarrow Y)$, $\mu Y.\bullet(X \rightarrow Y)$, $X \rightarrow \top$ and $\mu Y.X \rightarrow \bullet Y$ are proper in X , and neither X , $X \rightarrow Y$ nor $\mu Y.\mu Z.X \rightarrow Y$ is proper in X . Obviously, the following proposition holds.

Proposition 2.5. *If A is a \top -variant, then A is proper in every X .*

We can now define the syntax of type expressions of $\lambda\mathbf{A}$. A type expression of $\lambda\mathbf{A}$ is a pseudo type expression such that A is proper in X for any of its subexpressions in the form of $\mu X.A$. We denote the set of well-formed type expressions by **TExp**, which, more precisely, can be defined as follows.

Definition 2.6 (Type expressions). We define the set **TExp** of *type expressions* to be the smallest set of pseudo type expressions that satisfy

1. $X \in \mathbf{TExp}$ for every type variable X .
2. If $A \in \mathbf{TExp}$ and $B \in \mathbf{TExp}$, then $A \rightarrow B \in \mathbf{TExp}$.
3. If $A \in \mathbf{TExp}$, then $\bullet A \in \mathbf{TExp}$.
4. If $A \in \mathbf{TExp}$ and A is proper in X , then $\mu X.A \in \mathbf{TExp}$.

For example, X , $X \rightarrow Y$, $(\mu X.\bullet X \rightarrow Y) \rightarrow Z$, $\mu X.X \rightarrow \top$ and $\mu X.\bullet \mu Y.X \rightarrow Z$ are type expressions, and neither $\mu X.X \rightarrow Y$ nor $\mu X.\mu Y.X \rightarrow Y$ is a type expression.

When we consider free type variables occurring in a type expression, we can ignore those occurring in \top -variants, because they do not affect the meaning of the type expression. We define two sets of type variables that effectively occur in a type expression. One consists of those that occur at positive positions, and the other at negative positions.

Definition 2.7. Let A be a (pseudo) type expression. We define two sets $ETV^+(A)$ and $ETV^-(A)$ of type variables as follows.

$$\begin{aligned}
ETV^\pm(A) &= \{\} && (A \text{ is a } \top\text{-variant}) \\
ETV^+(X) &= \{X\} \\
ETV^-(X) &= \{\} \\
ETV^\pm(\bullet A) &= ETV^\pm(A) && (\bullet A \text{ is not a } \top\text{-variant}) \\
ETV^\pm(A \rightarrow B) &= ETV^\mp(A) \cup ETV^\pm(B) && (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
ETV^\pm(\mu X.A) &= \begin{cases} (ETV^\pm(A) \cup ETV^\mp(A)) - \{X\} & \left(\begin{array}{l} \mu X.A \text{ is not a } \top\text{-variant} \\ \text{and } X \in ETV^-(A) \end{array} \right) \\ ETV^\pm(A) - \{X\} & \left(\begin{array}{l} \mu X.A \text{ is not a } \top\text{-variant} \\ \text{and } X \notin ETV^-(A) \end{array} \right) \end{cases}
\end{aligned}$$

We also define $ETV(A)$ as $ETV(A) = ETV^+(A) \cup ETV^-(A)$.

For example, $ETV^+(\mu X.\bullet(X \rightarrow Y) \rightarrow Z) = \{Z\}$, $ETV^-(\mu X.\bullet(X \rightarrow Y) \rightarrow Z) = \{Y\}$, $ETV^\pm(\mu X.(Y \rightarrow Z) \rightarrow \bullet X) = \{\}$, and $ETV^\pm(\mu X.\bullet(X \rightarrow Y \rightarrow Z)) = \{Y, Z\}$. Obviously, $ETV^\pm(A) \subseteq FTV(A)$. We can easily check that α -conversion does not affect the definition of $ETV^\pm(A)$ by Proposition 2.3.5.

Some propositions in the present paper will be proved by induction on the following two kinds of structures of type expressions. The height $h(A)$ just reflects the syntactical one and the rank $r(A)$ does the semantic one in a fixed world.

Definition 2.8. Let A be a (pseudo) type expression. We define $h(A)$, the *height* of A , and $r(A)$, the *rank* of A , as follows.

$$\begin{aligned}
h(X) &= 0 \\
h(\bullet A) &= h(A) + 1 \\
h(A \rightarrow B) &= \max(h(A), h(B)) + 1 \\
h(\mu X.A) &= h(A) + 1 \\
r(A) &= 0 && (A \text{ is a } \top\text{-variant}) \\
r(X) &= 0 \\
r(\bullet A) &= 0 && (\bullet A \text{ is not a } \top\text{-variant}) \\
r(A \rightarrow B) &= \max(r(A), r(B)) + 1 && (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
r(\mu X.A) &= r(A) + 1 && (\mu X.A \text{ is not a } \top\text{-variant})
\end{aligned}$$

Proposition 2.9. Let A and B be (pseudo) type expressions.

1. $A[B/X]^t = A^t[B^t/X]$.
2. If A is proper in X , then so is A^t .
3. If A is a type expression, then so is A^t .
4. $ETV^-(A^t) = \{\}$ and $ETV^+(A^t) \subseteq ETV^+(A)$.
5. A is a \top -variant if and only if $ETV(A) = \{\}$, provided that $A \in \mathbf{TExp}$.

Proof. By straightforward induction on $h(A)$, and by cases on the form of A using Propositions 2.3 and 2.5. Use Item 2 for 3, and use Items 3 and 4 for 5. \square

In the rest of the present paper, we use A, B, C, \dots to denote only type expressions. Type expressions also have the following basic properties concerning \top -variants and properness.

Proposition 2.10. 1. If A is a \top -variant, then so is $A[B/X]$.

2. If $A[B/X]$ is a \top -variant, then (a) A is already a \top -variant, or (b) $X \in ETV^+(A^t)$ and B is a \top -variant.
3. If $A[B/X]$ is a \top -variant and A is proper in X , then $\mu X.A$ is also a \top -variant.
4. $\mu X.A$ is a \top -variant if and only if so is $A[\mu X.A/X]$.

Proof. Let $A^t = \bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} Z$, where $Y_i \notin FTV(B) \cup \{X\}$ for every i . By Proposition 2.9.1,

$$A[B/X]^t = \begin{cases} A^t & (Z \neq X) \\ \bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} B^t & (Z = X). \end{cases}$$

Therefore, we get Item 1 by Proposition 2.3.4 since $Z \neq X$ if A is a \top -variant, and similarly Item 2 since $Z = X$ implies $X \in ETV^+(A^t)$. We also get Item 3 since $(\mu X.A)^t$ is either $\mu X.(A[B/X])^t$ or $\mu X. \bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} X$. The “if part” of Item 4 follows 3, and the “only if” part is also straightforward since $A[\mu X.A/X]^t$ is either A^t (in case of $Z \neq X$), or $\bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} \mu X.A^t$ (in case of $Z = X$). \square

Proposition 2.11. 1. If $X \in ETV^\pm(A)$, $X \neq Y$ and B is not a \top -variant, then $X \in ETV^\pm(A[B/Y])$.

2. If $X \notin ETV^\pm(A)$ and $X \notin ETV(B)$, then $X \notin ETV^\pm(A[B/Y])$.

Proof. By straightforward induction on $h(A)$, and by cases on the form of A . For Item 1, note that A is not a \top -variant by Proposition 2.9.5 since $ETV(A) \neq \{\}$. Furthermore, $A[B/Y]$ is not a \top -variant either, by Proposition 2.10.2. Similarly, for Item 2, we only have to handle the case that neither A nor $A[B/Y]$ is a \top -variant, because $ETV^\pm(A[B/Y]) = \{\}$ by Proposition 2.9.5 if $A[B/Y]$ is a \top -variant, and because $A[B/Y]$ is a \top -variant by Proposition 2.10.1 if so is A . \square

- Proposition 2.12.** 1. If $X \in ETV^+(A)$ and $Y \in ETV^\pm(B)$, then $Y \in ETV^\pm(A[B/X])$.
 2. If $X \in ETV^-(A)$ and $Y \in ETV^\pm(B)$, then $Y \in ETV^\mp(A[B/X])$.
 3. If $X \notin ETV^+(A)$ and $Y \notin ETV^\pm(A) \cup ETV^\mp(B)$, then $Y \notin ETV^\pm(A[B/X])$.
 4. If $X \notin ETV^-(A)$ and $Y \notin ETV^\pm(A) \cup ETV^\pm(B)$, then $Y \notin ETV^\pm(A[B/X])$.

Proof. By simultaneous induction on $h(A)$, and by cases on the form of A . For Items 1 and 2, note that neither A , B nor $A[B/X]$ is a \top -variant by Propositions 2.9.5 and 2.10.2. In case of $A = \mu Z.C$ for some Z and C , we use Proposition 2.11.1 to show that $Z \in ETV^-(C)$ implies $Z \in ETV^-(C[B/X])$. Similarly, for Items 3 and 4, by Propositions 2.9.5 and 2.10.1, it suffices to consider the case that neither A nor $A[B/X]$ is a \top -variant. In case of $A = \mu Z.C$ for some Z and C , we use Proposition 2.11.2 to show that $Z \notin ETV^-(C)$ implies $Z \notin ETV^-(C[B/X])$. \square

- Proposition 2.13.** 1. If A is proper in X , then $r(A[B/X]) \leq r(A)$ for every B .
 2. If $\mu X.A$ is not a \top -variant, then $r(A[\mu X.A/X]) < r(\mu X.A)$.
 3. If A and B are proper in X , then $A[B/Y]$ is also proper in X for any Y .
 4. If A is proper in X , then so is $A[B/X]$ for every B .

Proof. By straightforward induction on $h(A)$, and by cases on the form of A using Proposition 2.10.1, where Item 2 immediately follows from 1. Note that every \top -variant is proper in any type variable, and that $r(A[B/X]) = 0$ for every \top -variant A . Hence, it suffices to only consider the case that A is not a \top -variant. \square

Definition 2.14. Let X be a type variable, and A a type expression. The *positive \bullet -depth* $dp_\bullet^+(A, X)$ and the *negative \bullet -depth* $dp_\bullet^-(A, X)$ of X in A are defined as follows.

$$\begin{aligned}
 dp_\bullet^\pm(A, X) &= \infty && (A \text{ is a } \top\text{-variant}) \\
 dp_\bullet^+(X, X) &= 0 \\
 dp_\bullet^-(X, X) &= \infty \\
 dp_\bullet^\pm(Y, X) &= \infty && (X \neq Y) \\
 dp_\bullet^\pm(\bullet A, X) &= dp_\bullet^\pm(A, X) + 1 && (\bullet A \text{ is not a } \top\text{-variant}) \\
 dp_\bullet^\pm(A \rightarrow B, X) &= \min(dp_\bullet^\mp(A, X), dp_\bullet^\pm(B, X)) && (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
 dp_\bullet^\pm(\mu Y.A, X) &= \min(dp_\bullet^\pm(A, X), dp_\bullet^\mp(A, Y) + dp_\bullet^\mp(A, X)) && (X \neq Y \text{ and } \mu Y.A \text{ is not a } \top\text{-variant})
 \end{aligned}$$

Similarly, the *positive (respectively, negative) \rightarrow -depth* $dp_{\rightarrow}^\pm(A, X)$ of X in A is defined as follows.

$$\begin{aligned}
 dp_{\rightarrow}^\pm(A, X) &= \infty && (A \text{ is a } \top\text{-variant}) \\
 dp_{\rightarrow}^+(X, X) &= 0 \\
 dp_{\rightarrow}^-(X, X) &= \infty \\
 dp_{\rightarrow}^\pm(Y, X) &= \infty && (X \neq Y) \\
 dp_{\rightarrow}^\pm(\bullet A, X) &= dp_{\rightarrow}^\pm(A, X) && (\bullet A \text{ is not a } \top\text{-variant}) \\
 dp_{\rightarrow}^\pm(A \rightarrow B, X) &= \min(dp_{\rightarrow}^\mp(A, X), dp_{\rightarrow}^\pm(B, X)) + 1 && (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
 dp_{\rightarrow}^\pm(\mu Y.A, X) &= \min(dp_{\rightarrow}^\pm(A, X), dp_{\rightarrow}^\mp(A, Y) + dp_{\rightarrow}^\mp(A, X)) && (X \neq Y \text{ and } \mu Y.A \text{ is not a } \top\text{-variant})
 \end{aligned}$$

The domains of $+$ and \min are assumed to be naturally extended to $\{0, 1, 2, \dots, \infty\}$. It can be easily verified that α -conversion does not affect the definition of $dp_\bullet^\pm(A, X)$ or $dp_{\rightarrow}^\pm(A, X)$. We also define the *\bullet -depth* $dp_\bullet(A, X)$ and *\rightarrow -depth* $dp_{\rightarrow}(A, X)$ of X in A as follows.

$$\begin{aligned}
 dp_\bullet(A, X) &= \min(dp_\bullet^+(A, X), dp_\bullet^-(A, X)) \\
 dp_{\rightarrow}(A, X) &= \min(dp_{\rightarrow}^+(A, X), dp_{\rightarrow}^-(A, X))
 \end{aligned}$$

For example, let $A = \mu X. \bullet(X \rightarrow \bullet Y) \rightarrow Z$ and $B = \mu X. \bullet(X \rightarrow Y \rightarrow Z)$. Then, $dp_{\bullet}^+(A, Y) = dp_{\rightarrow}^+(A, Y) = 2$, $dp_{\bullet}^-(A, Y) = dp_{\rightarrow}^-(A, Y) = \infty$, $dp_{\bullet}^+(B, Y) = 2$, $dp_{\rightarrow}^+(B, Y) = 3$, $dp_{\bullet}^-(B, Y) = 1$, and $dp_{\rightarrow}^-(B, Y) = 2$.

Proposition 2.15. *Let dp be either dp_{\bullet} or dp_{\rightarrow} , and suppose that B is not a \top -variant.*

1. $dp^{\pm}(A, X) < \infty$ if and only if $X \in ETV^{\pm}(A)$.
2. If $X \neq Y$ and $Y \notin ETV(B)$, then $dp^{\pm}(A[B/X], Y) = dp^{\pm}(A, Y)$.
3. If $X \neq Y$, then $dp^{\pm}(A[B/X], Y) = \min(dp^{\pm}(A, Y), dp^+(A, X) + dp^{\pm}(B, Y), dp^-(A, X) + dp^{\mp}(B, Y))$.
4. $dp^{\pm}(A[B/X], X) = \min(dp^+(A, X) + dp^{\pm}(B, X), dp^-(A, X) + dp^{\mp}(B, X))$.
5. A is proper in X if and only if $dp_{\bullet}(A, X) > 0$.
6. If $X \notin ETV(A)$, then A is proper in X .
7. $dp_{\rightarrow}^-(A, X) > 0$.
8. If $dp_{\rightarrow}^+(A, X) = 0$, then $A = \bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} X$ for some $n, m_0, m_1, m_2, \dots, m_n, Y_1, Y_2, \dots, Y_n$ such that $X \neq Y_i$ for any $i \in \{1, 2, \dots, n\}$.
9. If A is proper in X and $\mu X. A$ is not a \top -variant, then $dp_{\rightarrow}^+(A, X) > 0$.

Proof. For each item, the proof proceeds by induction on $h(A)$, and by cases on the form of A . Note that all the statements are almost trivial if A is a \top -variant. Use Item 1, Propositions 2.10.1 and 2.10.2 for Items 2, 3 and 4. Use also Item 3 for 4. Use Items 1, 7 and Proposition 2.9.5 for Item 8, from which Item 9 immediately follows. \square

3. Semantics of types

In this section, we define the semantics of type expressions. To fix notation, we start with some necessary definitions about the standard untyped λ -calculus.

Definition 3.1 (Untyped λ -terms). The syntax of the λ -terms is defined relatively to a set **Var** of countably infinite *individual variables* (f, g, h, x, y, z, \dots). The set **Exp** of λ -terms is defined by the following BNF notation.

$$\begin{array}{ll} \mathbf{Exp} ::= \mathbf{Var} & (\text{individual variables}) \\ | \lambda \mathbf{Var}. \mathbf{Exp} & (\lambda\text{-abstractions}) \\ | \mathbf{Exp} \mathbf{Exp} & (\text{applications}) \end{array}$$

We use M, N, K, L, \dots to denote λ -terms. Free and bound occurrences of individual variables and the notion of α -convertibility are defined in the standard manner. Hereafter, we identify λ -terms by this α -convertibility. We denote the set of individual variables occurring freely in M by $FV(M)$, and use $M[N_1/x_1, \dots, N_n/x_n]$ to denote the λ -term obtained from a λ -term M by substituting N_1, \dots, N_n for each free occurrence of individual variables x_1, \dots, x_n , respectively, with necessary α -conversion to avoid accidental capture of free variables.

Definition 3.2 (β -reduction). Let the syntax of context $\mathcal{C}[]$ of λ -term be defined in the standard way as follows.

$$\mathcal{C}[] ::= [] \mid \lambda \mathbf{Var}. \mathcal{C}[] \mid \mathcal{C}[] \mathbf{Exp} \mid \mathbf{Exp} \mathcal{C}[]$$

The standard notion of β -reduction, a binary relation \rightarrow_{β} over **Exp**, is defined by

$$\mathcal{C}[(\lambda x. M)N] \rightarrow_{\beta} \mathcal{C}[M[N/x]],$$

where \mathcal{C} is an arbitrary context of λ -term.

We denote the transitive and reflexive closure of \rightarrow_{β} by $\xrightarrow{*}_{\beta}$, and the symmetric closure of \rightarrow_{β} by \leftrightarrow_{β} . We define the equivalence relation \equiv_{β} to be the transitive and reflexive closure of \leftrightarrow_{β} .

Definition 3.3. Let ρ be a mapping from a set T to a set S , and let $x \in T$ and $v \in S$. We define a mapping $\rho[v/x]$ by

$$\rho[v/x](y) = \begin{cases} v & (y = x) \\ \rho(y) & (y \neq x). \end{cases}$$

Throughout the present paper, let $\langle \mathcal{V}, \cdot, \llbracket \cdot \rrbracket^{\mathcal{V}} \rangle$ be a syntactical λ -algebra of the untyped λ -calculus. Each λ -term M is interpreted as an element of \mathcal{V} , which is denoted by $\llbracket M \rrbracket_{\rho}^{\mathcal{V}}$, where ρ is an *individual environment* that assigns an element of \mathcal{V} to each individual variable. We define syntactical λ -algebra in the standard way as follows [3].

Definition 3.4 (syntactical λ -algebra). A syntactical λ -algebra of **Exp** is a tuple $\langle \mathcal{V}, \cdot, \llbracket \cdot \rrbracket^{\mathcal{V}} \rangle$ such that

1. \mathcal{V} : a non-empty set.
2. $\cdot : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$.
3. $\llbracket - \rrbracket_{-}^{\mathcal{V}} : \mathbf{Exp} \rightarrow (\mathbf{Var} \rightarrow \mathcal{V}) \rightarrow \mathcal{V}$.
4. $\llbracket x \rrbracket_{\rho}^{\mathcal{V}} = \rho(x)$.
5. $\llbracket MN \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M \rrbracket_{\rho}^{\mathcal{V}} \cdot \llbracket N \rrbracket_{\rho}^{\mathcal{V}}$.
6. $\llbracket \lambda x. M \rrbracket_{\rho}^{\mathcal{V}} \cdot v = \llbracket M \rrbracket_{\rho[v/x]}^{\mathcal{V}}$.
7. If $\rho(x) = \rho'(x)$ for every $x \in FV(M)$, then $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M \rrbracket_{\rho'}^{\mathcal{V}}$.
8. If $M \equiv_{\beta} N$, then $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} = \llbracket N \rrbracket_{\rho}^{\mathcal{V}}$.

Proposition 3.5. $\llbracket M[N/x] \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M \rrbracket_{\rho[\llbracket N \rrbracket_{\rho}^{\mathcal{V}}/x]}^{\mathcal{V}}$.

Proof. Since $M[N/x] \equiv_{\beta} (\lambda y. M)N$, $\llbracket M[N/x] \rrbracket_{\rho}^{\mathcal{V}} = \llbracket (\lambda y. M) N \rrbracket_{\rho}^{\mathcal{V}} = \llbracket \lambda y. M \rrbracket_{\rho}^{\mathcal{V}} \cdot \llbracket N \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M \rrbracket_{\rho[\llbracket N \rrbracket_{\rho}^{\mathcal{V}}/x]}^{\mathcal{V}}$. \square

Roughly, every type expression is interpreted as a subset of \mathcal{V} of a syntactical λ -algebra in each possible world of a Kripke frame, where we consider the following two classes of frames.

Definition 3.6. A *well-founded* frame is a pair $\langle \mathcal{W}, \triangleright \rangle$, which consists of a non-empty set \mathcal{W} of *possible worlds* and an *accessibility relation* \triangleright on \mathcal{W} such that

1. The relation \triangleright is (conversely) well-founded, i.e., there is no infinite sequence such that $p_0 \triangleright p_1 \triangleright p_2 \triangleright p_3 \triangleright \dots$.

We say the accessibility relation \triangleright is *locally linear* if and only if it satisfies the following additional condition, and define $\lambda\mathbf{A}$ -frames as well-founded frames whose accessibility relation is locally linear.

2. If $p \triangleright q$, then $p \triangleright^* r \triangleright q$ for some r such that $r \triangleright s$ implies $q \triangleright^* s$ for any s , where \triangleright^* denotes the reflexive and transitive closure of \triangleright .

The additional condition says that the accessibility relations is not *locally, in a sense*, branching. Any well-founded and linear binary relation \triangleright satisfies this condition². For example, the set of non-negative integers,

²It suffices to let r be a minimal element such that $r \triangleright q$.

or ordinals, and the “greater than” relation $>$ constitutes a $\lambda\mathbf{A}$ -frame. This condition does not always mean that the accessibility relation is not branching. For example, the following is a $\lambda\mathbf{A}$ -frame, but branching.

$$\begin{aligned}\mathcal{W} &= \{ \langle n, m \rangle \mid n \in \{0, 1, 2\} \text{ and } m \in \mathbb{N} \} \\ \langle n, m \rangle \triangleright \langle n', m' \rangle &\text{ iff } \begin{cases} n = 0 \text{ and } n' \in \{1, 2\}, \text{ or} \\ n = n' \text{ and } m > m' \end{cases}\end{aligned}$$

Note that we do not impose that \triangleright is transitive. It will be shown that our interpretation of types does not depend on whether \triangleright is transitive or not (Proposition 3.11).

Definition 3.7. A mapping t from \mathcal{W} to the power set $\mathcal{P}(\mathcal{V})$ of \mathcal{V} is said to be *hereditary* if and only if

$$p \triangleright q \text{ implies } t(p) \subseteq t(q).$$

A mapping η from $\mathbf{TVar} \times \mathcal{W}$ to $\mathcal{P}(\mathcal{V})$ is called a *type environment*, and also said to be *hereditary* if and only if

$$p \triangleright q \text{ implies } \eta(X, p) \subseteq \eta(X, q) \text{ for any } X \in \mathbf{TVar}.$$

In this paper, we only consider hereditary type environments. Given a well-founded frame and a hereditary type environment η , each type expression A is interpreted as a hereditary mapping from \mathcal{W} to $\mathcal{P}(\mathcal{V})$ as follows.

Definition 3.8 (Semantics of types). Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame, and η a hereditary type environment. We define a hereditary mapping $\mathcal{I}(A)^\eta$ from \mathcal{W} to $\mathcal{P}(\mathcal{V})$ for each type expression A by extending η as follows, where we prefer to write $\eta(X)_p$ and $\mathcal{I}(A)_p^\eta$ rather than $\eta(X, p)$ and $\mathcal{I}(A)^\eta(p)$, respectively.

$$\begin{aligned}\mathcal{I}(A)_p^\eta &= \mathcal{V} && (A \text{ is a } \top\text{-variant}) \\ \mathcal{I}(X)_p^\eta &= \eta(X)_p \\ \mathcal{I}(\bullet A)_p^\eta &= \{ u \mid \text{if } p \triangleright q, \text{ then } u \in \mathcal{I}(A)_q^\eta \} && (\bullet A \text{ is not a } \top\text{-variant}) \\ \mathcal{I}(A \rightarrow B)_p^\eta &= \{ u \mid \text{if } p \triangleright^* q, \text{ then } u \cdot v \in \mathcal{I}(B)_q^\eta \text{ for every } v \in \mathcal{I}(A)_q^\eta \} && (A \rightarrow B \text{ is not a } \top\text{-variant}) \\ \mathcal{I}(\mu X.A)_p^\eta &= \mathcal{I}(A[\mu X.A/X])_p^\eta && (\mu X.A \text{ is not a } \top\text{-variant})\end{aligned}$$

Note that the $\mathcal{I}(A)_p^\eta$ has been defined by induction on the lexicographic ordering of $\langle p, r(A) \rangle$, where we consider the transitive closure of \triangleright for the ordering of p , which is also well-founded since so is \triangleright . Because $r(A[\mu X.A/X]) < r(\mu X.A)$ by Proposition 2.13.2 when $\mu X.A$ is not a \top -variant, $\mathcal{I}(\mu X.A)_p^\eta$ is well defined. We can easily verify the following propositions.

Proposition 3.9. *The equations other than the first one in Definition 3.8 hold for any type expression, that is, the following equations hold.*

$$\begin{aligned}\mathcal{I}(X)_p^\eta &= \eta(X)_p \\ \mathcal{I}(\bullet A)_p^\eta &= \{ u \mid \text{if } p \triangleright q, \text{ then } u \in \mathcal{I}(A)_q^\eta \} \\ \mathcal{I}(A \rightarrow B)_p^\eta &= \{ u \mid \text{if } p \triangleright^* q, \text{ then } u \cdot v \in \mathcal{I}(B)_q^\eta \text{ for every } v \in \mathcal{I}(A)_q^\eta \} \\ \mathcal{I}(\mu X.A)_p^\eta &= \mathcal{I}(A[\mu X.A/X])_p^\eta\end{aligned}$$

Proof. Straightforward by Propositions 2.3 and 2.10.4. □

Proposition 3.10. *Let η be a hereditary type environment.*

1. $\mathcal{I}(A[B/X])_p^\eta = \mathcal{I}(A)_p^{\eta[\mathcal{I}(B)^\eta/X]}$.

2. $\mathcal{I}(A)^\eta$ is a hereditary mapping from \mathcal{W} to $\mathcal{P}(\mathcal{V})$; that is, $p \triangleright q$ implies $\mathcal{I}(A)_p^\eta \subseteq \mathcal{I}(A)_q^\eta$.

Proof. By straightforward induction on the lexicographic ordering of $\langle p, r(A) \rangle$, and by cases on the form of A . If A is a \top -variant, then so is $A[B/X]$ by Proposition 2.10.1; and hence, $\mathcal{I}(A[B/X])_p^\eta = \mathcal{I}(A)_p^\eta[\mathcal{I}(B)^\eta/X] = \mathcal{I}(A)_p^\eta = \mathcal{I}(A)_q^\eta = \mathcal{V}$ by Definition 3.8. Therefore, it suffices to only handle the case that A is not. \square

Proposition 3.11. Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame, and \triangleright^+ the transitive closure of \triangleright .

1. The accessibility relation \triangleright^+ is locally linear if and only if so is \triangleright .
2. Let \mathcal{I} and \mathcal{I}^+ be the interpretations over $\langle \mathcal{W}, \triangleright \rangle$ and $\langle \mathcal{W}, \triangleright^+ \rangle$, respectively. $\mathcal{I}(A)_p^\eta = \mathcal{I}^+(A)_p^\eta$ for every hereditary type environment η and $p \in \mathcal{W}$.

Proof. Straightforward. Use Propositions 3.9 and 3.10.2 for Item 2. \square

Adopting such an interpretation of types over a well-founded frame, or a $\lambda\mathbf{A}$ -frame, we can verify the subtyping relations between $\bullet(A \rightarrow B)$ and $\bullet A \rightarrow \bullet B$ discussed in Section 1 as follows.

Proposition 3.12. Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame. Then, $\mathcal{I}(\bullet(A \rightarrow B))_p^\eta \subseteq \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$ for any A, B, η and $p \in \mathcal{W}$.

Proof. Suppose that $u \in \mathcal{I}(\bullet(A \rightarrow B))_p^\eta$. To show that $u \in \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$, suppose also that $p \stackrel{*}{\triangleright} r$ and $v \in \mathcal{I}(\bullet A)_r^\eta$. By Proposition 3.9, it suffices to show that $u \cdot v \in \mathcal{I}(B)_q^\eta$ for every $q \triangleleft r$. Since $u \in \mathcal{I}(\bullet(A \rightarrow B))_p^\eta$ and $p \stackrel{*}{\triangleright} r$, we get $u \in \mathcal{I}(\bullet(A \rightarrow B))_r^\eta$ by Proposition 3.10.2. Hence, if $r \triangleright q$, then $u \in \mathcal{I}(A \rightarrow B)_q^\eta$; and therefore, $u \cdot v \in \mathcal{I}(B)_q^\eta$ by Proposition 3.9 because $v \in \mathcal{I}(A)_q^\eta$ from $v \in \mathcal{I}(\bullet A)_r^\eta$. \square

Theorem 3.13. Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame. Then, $\mathcal{I}(\bullet(A \rightarrow B))_p^\eta = \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$ for any A, B , hereditary η and $p \in \mathcal{W}$ if and only if \triangleright is locally linear.

Proof. For the “if” part, suppose that \triangleright satisfies Condition 2 of Definition 3.6. We get $\mathcal{I}(\bullet(A \rightarrow B))_p^\eta \subseteq \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$ by Proposition 3.12. To show the opposite, suppose also that $u \in \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$, $p \triangleright q' \stackrel{*}{\triangleright} q$ and $v \in \mathcal{I}(A)_{q'}^\eta$. It suffices to show that $u \cdot v \in \mathcal{I}(B)_q^\eta$. Let p' be the element of \mathcal{W} such that $p \stackrel{*}{\triangleright} p' \triangleright q$. By Condition 2 of Definition 3.6, there exists some $r \in \mathcal{W}$ such that

$$p \stackrel{*}{\triangleright} r, \tag{1}$$

$$r \triangleright q, \text{ and} \tag{2}$$

$$r \triangleright s \text{ implies } q \stackrel{*}{\triangleright} s \text{ for any } s. \tag{3}$$

By Proposition 3.10.2, we get $u \in \mathcal{I}(\bullet A \rightarrow \bullet B)_r^\eta$ from $u \in \mathcal{I}(\bullet A \rightarrow \bullet B)_p^\eta$ and (1), and get $v \in \mathcal{I}(\bullet A)_r^\eta$ from $v \in \mathcal{I}(A)_{q'}^\eta$ and (3). Therefore, $u \cdot v \in \mathcal{I}(\bullet B)_r^\eta$ by Proposition 3.9. We now get $u \cdot v \in \mathcal{I}(B)_q^\eta$ from this and (2) by Proposition 3.9.

For the “only if” part, suppose that \triangleright does not satisfy Condition 2, i.e., there exist some $p, q \in \mathcal{W}$ such that $p \triangleright q$, and

$$\text{for every } r \in \mathcal{W}, \text{ if } p \stackrel{*}{\triangleright} r \triangleright q, \text{ then } r \triangleright s \text{ and } q \not\stackrel{*}{\triangleright} s \text{ for some } s \in \mathcal{W}. \tag{4}$$

Since $p \stackrel{*}{\triangleright} p \triangleright q$, there exists some s such that $p \triangleright s$ and $q \not\stackrel{*}{\triangleright} s$. Then, consider the hereditary type environment η defined as follows.

$$\eta(X)_t = \begin{cases} \mathcal{V} & (q \stackrel{*}{\triangleright} t) \\ \{\} & (q \not\stackrel{*}{\triangleright} t) \end{cases} \tag{5}$$

$$\eta(Y)_t = \begin{cases} \mathcal{V} & (q \triangleright t) \\ \{\} & (q \not\triangleright t) \end{cases} \tag{6}$$

We can see that $\mathcal{I}(\bullet(X \rightarrow Y))_p^\eta = \{\}$, while $\mathcal{I}(\bullet X \rightarrow \bullet Y)_p^\eta = \mathcal{V}$. In fact, $\mathcal{I}(X \rightarrow Y)_q^\eta = \{\}$, since $\mathcal{I}(X)_q^\eta = \eta(X)_q = \mathcal{V}$ and $\mathcal{I}(Y)_q^\eta = \eta(Y)_q = \{\}$ by (5) and (6), respectively; and hence, $\mathcal{I}(\bullet(X \rightarrow Y))_p^\eta = \{\}$ from $p \triangleright q$. Furthermore, $\mathcal{I}(\bullet X \rightarrow \bullet Y)_p^\eta = \mathcal{V}$ can be also shown as follows. Suppose that $p \triangleright^* t$ and $\mathcal{I}(\bullet X)_t^\eta \neq \{\}$, which means that $t \triangleright s$ implies $\eta(X)_s = \mathcal{V}$ for any $s \in \mathcal{W}$ by (5); that is,

$$t \triangleright s \text{ implies } q \triangleright^* s \text{ for any } s.$$

Hence, $t \not\triangleright q$ by (4); and therefore, $t \triangleright s$ implies $q \triangleright s$ for any s . Then, by (6),

$$t \triangleright s \text{ implies } \eta(Y)_s = \mathcal{V} \text{ for any } s.$$

We thus get $\mathcal{I}(\bullet Y)_t^\eta = \mathcal{V}$ from $\mathcal{I}(\bullet X)_t^\eta \neq \{\}$ and $p \triangleright^* t$. Hence, $\mathcal{I}(\bullet X \rightarrow \bullet Y)_p^\eta = \mathcal{V}$. \square

4. Equality of types

In Section 6, we will introduce a modal typing system $\lambda\mathbf{A}$, which respect the semantics of type expressions based on $\lambda\mathbf{A}$ -frames. In this section, we discuss formal derivability of equality between type expressions, which will be incorporated into the modal typing system.

Definition 4.1 (\cong). The equivalence relation \cong on \mathbf{TExp} is defined to be the smallest binary relation that satisfies the following:

- (\cong -reflex) $A \cong A$.
- (\cong -symm) If $A \cong B$, then $B \cong A$.
- (\cong -trans) If $A \cong B$ and $B \cong C$, then $A \cong C$.
- (\cong - \bullet) If $A \cong B$, then $\bullet A \cong \bullet B$.
- (\cong - \rightarrow) If $A \cong C$ and $B \cong D$, then $A \rightarrow B \cong C \rightarrow D$.
- (\cong - $\rightarrow\top$) $A \rightarrow \top \cong \top$.
- (\cong -fix) $\mu X.A \cong A[\mu X.A/X]$.
- (\cong -uniq) If $A \cong C[A/X]$ and C is proper in X , then $A \cong \mu X.C$.

For example, $\mu X.Y \rightarrow \bullet X \cong \top$ by (\cong -uniq) since $(Y \rightarrow \bullet X)[\top/X] \cong \top$ as follows.

$$\begin{aligned} (Y \rightarrow \bullet X)[\top/X] &= Y \rightarrow \bullet \mu X.\bullet X \\ &\cong Y \rightarrow \mu X.\bullet X && \text{(by } (\cong\text{-reflex}), (\cong\text{-fix}), (\cong\text{-symm}) \text{ and } (\cong\text{-}\rightarrow)) \\ &\cong \mu X.\bullet X && \text{(by } (\cong\text{-}\rightarrow\top)) \\ &= \top \end{aligned}$$

The intended meaning of $A \cong B$ is that the interpretations of A and B are identical in any world of any well-founded frames. We will see in Proposition 4.12.1 that the following rule is derivable from the above.

$A \cong B$ implies $\mu X.A \cong \mu X.B$, provided that A or B is proper in X .

Definition 4.2 (\simeq). We similarly define another equivalence relation \simeq by adding the following condition to the definition of \cong .

$$(\simeq\text{-K/L}) \quad \bullet(A \rightarrow B) \simeq \bullet A \rightarrow \bullet B$$

This rule reflects the equivalence shown in Theorem 3.13, and is only valid for $\lambda\mathbf{A}$ -frames. Noted that $A \cong B$ implies $A \simeq B$. Roughly, two type expressions are equivalent modulo \cong , if their (possibly infinite) type expressions obtained by indefinite unfolding recursive types occurring in them are identical modulo the rule (\cong - $\rightarrow\top$). Also, they are equivalent modulo \simeq , if their indefinite unfoldings are identical modulo the rules (\cong - $\rightarrow\top$) and (\simeq -K/L).

4.1. Basic properties of the equality

In this subsection, we discuss some basic properties of \cong and \simeq . Most of the results are common to both \cong and \simeq . In the sequel, let \sim denotes either \cong or \simeq .

- Proposition 4.3.** 1. $\top \sim \bullet \top$.
 2. $\top \sim \mu X. \bullet^n X$ for every $n \geq 1$.
 3. $A \sim \bullet^{n+1} A$ if and only if $A \sim \top$.

Proof. Straightforward by $(\cong\text{-fix})$ and $(\cong\text{-uniq})$. \square

In the succeeding sections, we occasionally use the following proposition in proofs without mention.

Proposition 4.4. If $A \sim B$ and $C \sim D$, then $A[C/X] \sim B[D/X]$.

Proof. By induction on the derivation of $A \sim B$, and by cases on the last rule applied in it. The readers should refer to Appendix A.1 for a detailed proof. \square

Before showing the soundness of \cong and \simeq with respect to the intended semantics of type expressions, we confirm that the equality preserves whether a type expression is a \top -variant or not, since some notions introduced so far, such as properness, dp_{\bullet}^{\pm} , dp_{\rightarrow}^{\pm} and ETV^{\pm} , are defined in a way where \top -variants are treated as an exceptional case.

Proposition 4.5. Suppose that $A \sim B$. Then, A is a \top -variant if and only if so is B .

Proof. By simultaneous induction on the derivation of $A \sim B$ with the claim

$$dp_{\bullet}^{+}(A^t, Z) = dp_{\bullet}^{+}(B^t, Z) \text{ for any } Z,$$

and by cases on the last rule used in the derivation. Note that, by Propositions 2.9.4 and 2.15.1,

$$dp_{\bullet}^{-}(A^t, Z) = dp_{\bullet}^{-}(B^t, Z) = \infty \text{ for any } Z.$$

Use Propositions 2.3 and 2.10.4. The cases other than $(\cong\text{-fix})$ or $(\cong\text{-uniq})$ are straightforward.

In case of $(\cong\text{-fix})$, there exist some X and C such that $A = \mu X.C$ and $B = C[A/X]$. First, A is a \top -variant if and only if so is B by Proposition 2.10.4. Therefore, if either A or B is a \top -variant, so are A^t and B^t by Proposition 2.3.4; and hence, $dp_{\bullet}^{+}(A^t, Z) = dp_{\bullet}^{+}(B^t, Z) = \infty$ by Definition 2.14. On the other hand, if neither A nor B is a \top -variant, then assuming $X \neq Z$ without loss of generality,

$$\begin{aligned} dp_{\bullet}^{+}(B^t, Z) &= dp_{\bullet}^{+}(C^t[A/X], Z) && \text{(by Proposition 2.9.1)} \\ &= \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{+}(C^t, X) + dp_{\bullet}^{+}(A^t, Z), dp_{\bullet}^{-}(C^t, X) + dp_{\bullet}^{-}(A^t, Z)) \\ &&& \text{(by Proposition 2.15.3)} \\ &= \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{+}(C^t, X) + dp_{\bullet}^{+}(A^t, Z)) && \text{(since } dp_{\bullet}^{-}(C^t, X) = dp_{\bullet}^{-}(A^t, Z) = \infty) \\ &= \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{+}(C^t, X) + \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{-}(C^t, X) + dp_{\bullet}^{-}(C^t, Z))) \\ &&& \text{(by Definition 2.14)} \\ &= \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{+}(C^t, X) + dp_{\bullet}^{+}(C^t, Z)) && \text{(since } dp_{\bullet}^{-}(C^t, X) = dp_{\bullet}^{-}(C^t, Z) = \infty) \\ &= dp_{\bullet}^{+}(C^t, Z) \\ &= \min(dp_{\bullet}^{+}(C^t, Z), dp_{\bullet}^{-}(C^t, X) + dp_{\bullet}^{-}(C^t, Z)) && \text{(since } dp_{\bullet}^{-}(C^t, X) = dp_{\bullet}^{-}(C^t, Z) = \infty) \\ &= dp_{\bullet}^{+}(A^t, Z) && \text{(by Definition 2.14).} \end{aligned}$$

In case of $(\cong\text{-uniq})$, there exist some X and C such that $B = \mu X.C$, $A \sim C[A/X]$, and C is proper in X . By induction hypothesis, we have

$$A \text{ is a } \top\text{-variant iff so is } C[A/X], \text{ and} \tag{7}$$

$$dp_{\bullet}^{+}(A^t, Z) = dp_{\bullet}^{+}(C[A/X]^t, Z) \text{ for any } Z. \tag{8}$$

First, if A is a \top -variant, then so is $C[A/X]$ by (7); and therefore, so is $\mu X.C$ by Proposition 2.10.3. To show the converse, assume that $\mu X.C$ is a \top -variant whereas A is not. We show that this assumption leads to a contradiction. Let $C^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} Y$, and $A^t = \bullet^{m'_0} \mu X'_1. \bullet^{m'_1} \mu X'_2. \bullet^{m'_2} \dots \mu X'_{n'}. \bullet^{m'_{n'}} Y'$. Note that $Y \in \{X, X_1, X_2, \dots, X_n\}$ and $Y' \notin \{X'_1, X'_2, \dots, X'_{n'}\}$ since $\mu X.C$ is a \top -variant and A is not. Furthermore, since A is not, $C[A/X]^t$ is not a \top -variant either, by (7) and Proposition 2.3.4. Therefore, C^t is not a \top -variant either, by Propositions 2.9.1 and 2.10.1; and hence, $Y = X$ and $Y \neq X_i$ for every i . Note also that $m_0 + m_1 + \dots + m_n > 0$ by Propositions 2.9.2 and 2.15.5 since C is proper in X . Hence,

$$\begin{aligned} dp_{\bullet}^+(A^t, Y') &= dp_{\bullet}^+(C[A/X]^t, Y') && \text{(by (8))} \\ &= dp_{\bullet}^+(C^t[A^t/X], Y') && \text{(by Proposition 2.9.1)} \\ &= dp_{\bullet}^+(\bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} A^t, Y') \\ &= dp_{\bullet}^+(A^t, Y') + m_0 + m_1 + \dots + m_n && \text{(by Definition 2.14).} \end{aligned}$$

However, this is impossible since $dp_{\bullet}^+(A^t, Y') = m'_0 + m'_1 + \dots + m'_{n'} < \infty$ and $m_0 + m_1 + \dots + m_n > 0$. This completes the proof of the first part. For the second part, we assume that neither A^t nor $(\mu X.C)^t$ is a \top -variant, since $dp_{\bullet}^+(A^t, Z) = dp_{\bullet}^+((\mu X.C)^t, Z) = \infty$ by Definition 2.14 if they are. Note that C^t is also proper in X by Proposition 2.9.2; and hence, $dp_{\bullet}(C^t, X) > 0$ by Proposition 2.15.5. Then, assuming that $X \neq Z$ without loss of generality,

$$\begin{aligned} dp_{\bullet}^+(A^t, Z) &= dp_{\bullet}^+(C[A/X]^t, Z) && \text{(by (8))} \\ &= dp_{\bullet}^+(C^t[A^t/X], Z) && \text{(by Proposition 2.9.1)} \\ &= \min(dp_{\bullet}^+(C^t, Z), dp_{\bullet}^+(C^t, X) + dp_{\bullet}^+(A^t, Z), dp_{\bullet}^-(C^t, X) + dp_{\bullet}^-(A^t, Z)) \\ &&& \text{(by Proposition 2.15.3)} \\ &= \min(dp_{\bullet}^+(C^t, Z), dp_{\bullet}^+(C^t, X) + dp_{\bullet}^+(A^t, Z)) && \text{(since } dp_{\bullet}^-(C^t, X) = dp_{\bullet}^-(A^t, Z) = \infty) \\ &= dp_{\bullet}^+(C^t, Z) && \text{(since } dp_{\bullet}^+(C^t, X) > 0; \text{ valid even if } dp_{\bullet}^+(A^t, Z) = \infty) \\ &= \min(dp_{\bullet}^+(C^t, Z), dp_{\bullet}^-(C^t, X) + dp_{\bullet}^-(C^t, Z)) && \text{(since } dp_{\bullet}^-(C^t, X) = dp_{\bullet}^-(C^t, Z) = \infty) \\ &= dp_{\bullet}^+(\mu X.C^t, Z) && \text{(by Definition 2.14)} \\ &= dp_{\bullet}^+(B^t, Z) && \text{(by Definition 2.2).} \end{aligned} \quad \square$$

4.2. Soundness of the derivation rules for equality

We now proceed to show that the equivalence relations \cong and \simeq on type expressions well respect the semantics of types according to their intended frame classes. The last one step before the proof is to prove the following lemma, which says that if a type expression is proper in a type variable, then the interpretation of the type expression in a possible world does not depend on the interpretation of the type variable in that world.

Lemma 4.6. *Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame, A a type expression, and $p \in \mathcal{W}$. Let η and η' be hereditary type environments such that $\eta(X)_q = \eta'(X)_q$ for every $X \in \mathbf{TVar}$ and $q \in \mathcal{W}$ such that $p \triangleright q$. If for every X , either (a) A is proper in X , or (b) $\eta(X)_p = \eta'(X)_p$, then $\mathcal{I}(A)_p^\eta = \mathcal{I}(A)_p^{\eta'}$.*

Proof. By induction on the lexicographic ordering of $\langle p, r(A) \rangle$. Suppose that (a) or (b) holds for every $X \in \mathbf{TVar}$. If A is a \top -variant, then $\mathcal{I}(A)_p^\eta = \mathcal{I}(A)_p^{\eta'} = \mathcal{V}$ by Definition 3.8. Therefore, we show that $\mathcal{I}(A)_p^\eta = \mathcal{I}(A)_p^{\eta'}$ by cases on the form of A assuming that A is not a \top -variant.

Case: $A = Y$ for some Y . In this case, A is not proper in Y . Hence, $\mathcal{I}(A)_p^\eta = \eta(Y)_p = \eta'(Y)_p = \mathcal{I}(A)_p^{\eta'}$ from (b).

Case: $A = \bullet B$ for some B . Since $\eta(X)_q = \eta'(X)_q$ for every X and $q \triangleleft p$, we get $\mathcal{I}(B)_q^\eta = \mathcal{I}(B)_q^{\eta'}$ for every $q \triangleleft p$ by induction hypothesis. Therefore, $\mathcal{I}(\bullet B)_p^\eta = \{u \mid u \in \mathcal{I}(B)_q^\eta \text{ for every } q \triangleleft p\} = \{u \mid u \in \mathcal{I}(B)_q^{\eta'} \text{ for every } q \triangleleft p\} = \mathcal{I}(\bullet B)_p^{\eta'}$ by Definition 3.8.

Case: $A = B \rightarrow C$ for some B and C . Note that C is not a \top -variant since A is not. Therefore, $r(B), r(C) < r(A)$, and (a) implies both B and C are also proper in X . Hence, by induction hypothesis, we get $\mathcal{I}(B)_q^\eta = \mathcal{I}(B)_q^{\eta'}$ and $\mathcal{I}(C)_q^\eta = \mathcal{I}(C)_q^{\eta'}$ for every q such that $p \triangleright^* q$. Therefore, $\mathcal{I}(B \rightarrow C)_p^\eta = \mathcal{I}(B \rightarrow C)_p^{\eta'}$ by Definition 3.8.

Case: $A = \mu Y.C$ for some Y and C . We can assume that $X \neq Y$ without loss of generality. Note that $r(C[\mu Y.C/Y]) < r(\mu Y.C)$ by Proposition 2.13.2, and that $C[\mu Y.C/Y]$ is proper in X if so is A , by Definition 2.4 and Proposition 2.13.3. Hence,

$$\begin{aligned} \mathcal{I}(\mu Y.C)_p^\eta &= \mathcal{I}(C[\mu Y.C/Y])_p^\eta && \text{(by Definition 3.8)} \\ &= \mathcal{I}(C[\mu Y.C/Y])_p^{\eta'} && \text{(by induction hypothesis)} \\ &= \mathcal{I}(\mu Y.C)_p^{\eta'} && \text{(by Definition 3.8).} \end{aligned} \quad \square$$

Theorem 4.7 (Soundness of \cong). *Let $\langle \mathcal{W}, \triangleright \rangle$ be a well-founded frame, and consider the interpretation in the frame. If $A \cong B$, then $\mathcal{I}(A)^\eta = \mathcal{I}(B)^\eta$ for any type environment η .*

Proof. By induction on the derivation of $A \cong B$, and by cases on the last rule in the derivation. Suppose that $A \cong B$. If either A or B is a \top -variant, then so are both by Proposition 4.5; and hence, $\mathcal{I}(A)_p^\eta = \mathcal{I}(B)_p^\eta = \top$ for every p by Definition 3.8. Therefore, we show that $\mathcal{I}(A)^\eta = \mathcal{I}(B)^\eta$ assuming that neither is a \top -variant.

Cases: (\cong -reflex), (\cong -symm) and (\cong -trans). Trivial.

Case: (\cong - \bullet). In this case, there exist some A' and B' such that $A = \bullet A'$, $B = \bullet B'$ and $A' \cong B'$. We have $\mathcal{I}(A')_q^\eta = \mathcal{I}(B')_q^\eta$ for every $q \in \mathcal{W}$ by induction hypothesis. Therefore, $\mathcal{I}(\bullet A')_p^\eta = \{ u \mid u \in \mathcal{I}(A')_q^\eta \text{ for every } q \triangleleft p \} = \{ u \mid u \in \mathcal{I}(B')_q^\eta \text{ for every } q \triangleleft p \} = \mathcal{I}(\bullet B')_p^\eta$ by Definition 3.8.

Case: (\cong - \rightarrow). Similar to the previous case.

Case: (\cong - $\rightarrow \top$). Impossible because we assumed that neither A nor B is a \top -variant.

Case: (\cong -fix). Obvious from Definition 3.8.

Case: (\cong -uniq). There exist some X and C such that $B = \mu X.C$, $A \cong C[A/X]$ and C is proper in X . By induction hypothesis,

$$\mathcal{I}(A)_p^{\eta'} = \mathcal{I}(C[A/X])_p^{\eta'} \text{ for every } \eta'. \quad (9)$$

We show that $\mathcal{I}(A)_p^\eta = \mathcal{I}(\mu X.C)_p^\eta$ for every $p \in \mathcal{W}$ by induction on p . The induction hypothesis in this induction is

$$\mathcal{I}(A)_q^\eta = \mathcal{I}(\mu X.C)_q^\eta \text{ for every } q \triangleleft p. \quad (10)$$

Therefore,

$$\begin{aligned} \mathcal{I}(A)_p^\eta &= \mathcal{I}(C[A/X])_p^\eta && \text{(by (9))} \\ &= \mathcal{I}(C)_p^{\eta[\mathcal{I}(A)^\eta/X]} && \text{(by Proposition 3.10.1)} \\ &= \mathcal{I}(C)_p^{\eta[\mathcal{I}(\mu X.C)^\eta/X]} && \text{(by (10) and Lemma 4.6)} \\ &= \mathcal{I}(C[\mu X.C/X])_p^\eta && \text{(by Proposition 3.10.1)} \\ &= \mathcal{I}(\mu X.C)_p^\eta && \text{(by Definition 3.8).} \end{aligned} \quad \square$$

Theorem 4.8 (Soundness of \simeq with respect to $\lambda\mathbf{A}$ -frames). *Let $\langle \mathcal{W}, \triangleright \rangle$ be a $\lambda\mathbf{A}$ -frame, η a hereditary type environment. If $A \simeq B$, then $\mathcal{I}(A)^\eta = \mathcal{I}(B)^\eta$.*

Proof. The proof is quite parallel to the one of Theorem 4.7. The only additional task is to check the rule (\simeq -K/L), which is straightforward from Theorem 3.13. \square

4.3. Properties of type expressions preserved by the equality

At the end of Subsection 4.1, we showed that the equality preserves whether a type expression is a \top -variant or not. In this subsection, we shall confirm the fact that the equivalence relations also preserve other basic properties of type expressions such as properness, dp_{\bullet}^{\pm} , dp_{\rightarrow}^{\pm} and ETV^{\pm} . Most of them are proved by induction on the derivation of the equality.

Proposition 4.9. $A \sim B$ implies $A^t \sim B^t$.

Proof. By induction on the derivation of $A \sim B$, and by cases on the last rule used in the derivation. We employ Proposition 2.9 in the case that last rule is $(\cong\text{-fix})$ or $(\cong\text{-uniq})$. \square

Proposition 4.10. If $A \sim B$, then $dp_{\bullet}^{\pm}(A, X) = dp_{\bullet}^{\pm}(B, X)$ and $dp_{\rightarrow}^{\pm}(A, X) = dp_{\rightarrow}^{\pm}(B, X)$.

Proof. By induction on the derivation of $A \sim B$, and by cases on the rule applied last. Suppose that $A \sim B$. The cases other than $(\cong\text{-fix})$ or $(\cong\text{-uniq})$ are again straightforward. Let dp be either dp_{\bullet} or dp_{\rightarrow} . If either A or B is a \top -variant, then so are they by Proposition 4.5; and therefore, $dp^{\pm}(A, X) = dp^{\pm}(B, X) = \infty$ by Definition 2.14. Hence, we assume that neither is. In case of $(\cong\text{-fix})$, there exist some Y and C such that $A = \mu Y.C$ and $B = C[A/Y]$, where we can assume that $Y \neq X$ without loss of generality. Note that $dp(C, Y) > 0$ by Propositions 2.15.5, 2.15.7 and 2.15.9, since C is proper in Y . Therefore,

$$\begin{aligned}
dp^{\pm}(B, X) &= dp^{\pm}(C[A/Y], X) \\
&= \min(dp^{\pm}(C, X), dp^{+}(C, Y) + dp^{\pm}(A, X), dp^{-}(C, Y) + dp^{\mp}(A, X)) \quad (\text{by Proposition 2.15.3}) \\
&= \min(dp^{\pm}(C, X), \\
&\quad dp^{+}(C, Y) + \min(dp^{\pm}(C, X), dp^{-}(C, Y) + dp^{\mp}(C, X)), \\
&\quad dp^{-}(C, Y) + \min(dp^{\mp}(C, X), dp^{-}(C, Y) + dp^{\pm}(C, X))) \quad (\text{by Definition 2.14}) \\
&= \min(dp^{\pm}(C, X), \\
&\quad dp^{+}(C, Y) + \min(dp^{\pm}(C, X), dp^{-}(C, Y) + dp^{\mp}(C, X)), \\
&\quad dp^{-}(C, Y) + dp^{\mp}(C, X)) \quad (\text{since } dp^{-}(C, Y) > 0) \\
&= \min(dp^{\pm}(C, X), dp^{-}(C, Y) + dp^{\mp}(C, X)) \quad (\text{since } dp^{+}(C, Y) > 0) \\
&= dp^{\pm}(\mu Y.C, X) \quad (\text{by Definition 2.14}).
\end{aligned}$$

In case of $(\cong\text{-uniq})$, there exist some Y and C such that $B = \mu Y.C$, $A \sim C[A/Y]$, and C is proper in Y , where we again assume that $Y \neq X$ without loss of generality. In this case, $C[A/Y]$ is not a \top -variant either, by Proposition 4.5. Note that $dp(C, Y) > 0$ by Propositions 2.15.5, 2.15.7 and 2.15.9, since C is proper in Y . Therefore,

$$\begin{aligned}
dp^{\pm}(A, X) &= dp^{\pm}(C[A/Y], X) \quad (\text{by induction hypothesis}) \\
&= \min(dp^{\pm}(C, X), dp^{+}(C, Y) + dp^{\pm}(A, X), dp^{-}(C, Y) + dp^{\mp}(A, X)) \quad (\text{by Proposition 2.15.3}) \\
&= \min(dp^{\pm}(C, X), dp^{-}(C, Y) + dp^{\mp}(A, X)) \quad (\text{since } dp^{+}(C, Y) > 0; \text{ valid even if } dp^{\pm}(A, X) = \infty) \\
&= \min(dp^{\pm}(C, X), dp^{-}(C, Y) + \min(dp^{\mp}(C, X), dp^{-}(C, Y) + dp^{\pm}(A, X))) \quad (\text{by the equation so far}) \\
&= \min(dp^{\pm}(C, X), dp^{-}(C, Y) + dp^{\mp}(C, X)) \quad (\text{since } dp^{-}(C, Y) > 0; \text{ valid even if } dp^{\pm}(A, X) = \infty) \\
&= dp^{\pm}(\mu Y.C, X) \quad (\text{by Definition 2.14}). \quad \square
\end{aligned}$$

Proposition 4.11. Suppose that $A \sim B$.

1. $ETV^{\pm}(A) = ETV^{\pm}(B)$.
2. A is proper in X if and only if so is B .

Proof. Straightforward by Propositions 2.15.1, 2.15.5 and 4.10. \square

By the facts obtained so far, we can show the following derived rules.

Proposition 4.12. 1. If $A \sim B$, then $\mu X.A \sim \mu X.B$.

2. $\mu X.A[X/Y] \sim \mu X.A[A[X/Y]/Y]$.

Proof. For Item 1, suppose that $A \sim B$. Then, $\mu X.A \sim A[\mu X.A/X] \sim B[\mu X.A/X]$ by $(\cong\text{-fix})$ and Proposition 4.4. Therefore, $\mu X.A \sim \mu X.B$ by $(\cong\text{-uniq})$. For Item 2, whether $X = Y$ or not,

$$\begin{aligned} \mu X.A[X/Y] &\sim A[X/Y][\mu X.A[X/Y]/X] && \text{(by } (\cong\text{-fix}) \text{)} \\ &\sim A[\mu X.A[X/Y]/Y][\mu X.A[X/Y]/X] && \text{(since } X \notin FTV(\mu X.A[X/Y]) \text{)} \\ &\sim A[A[X/Y][\mu X.A[X/Y]/X]/Y][\mu X.A[X/Y]/X] && \text{(by } (\cong\text{-fix}) \text{ and Proposition 4.4)} \\ &\sim A[A[X/Y]/Y][\mu X.A[X/Y]/X] && \text{(since } X \notin FTV(A[X/Y][\mu X.A[X/Y]/X]) \text{)}. \end{aligned}$$

Hence, $\mu X.A[X/Y] \sim \mu X.A[A[X/Y]/Y]$ by $(\cong\text{-uniq})$, again. \square

Proposition 4.5 says that the relation \sim preserves whether a type expression is a \top -variant or not. Furthermore, we can show that a type expression A is a \top -variant if and only if $A \sim \top$. It does not depend on which of \cong and \simeq we consider as the equality between types. Because it is syntactically decidable whether a type expression is a \top -variant or not, it is decidable whether $A \sim \top$ or not,

Lemma 4.13. Let $A^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} Y$ for some n , $m_0, m_1, m_2, \dots, m_n$, X_1, X_2, \dots, X_n and Y .

1. $A \sim \top$ if the following is the case.
 - (a) $Y = X_i$ for some i such that $Y \notin \{X_{i+1}, X_{i+2}, \dots, X_n\}$, and $m_i + m_{i+1} + m_{i+2} + \dots + m_n \geq 1$.
2. $A[\top/Y] \sim \top$ if the following is the case.
 - (b) $Y \notin \{X_1, X_2, \dots, X_n\}$.

Proof. By simultaneous induction on $h(A)$, and by cases on the form of A .

Case: $A = Z$ for some Z . In this case, $Y = Z$, $n = 0$ and $m_0 = 0$ since $A^t = Z$. Hence, (a) is not the case. The second item is trivial since $A[\top/Y] = Y[\top/Y] = \top$.

Case: $A = B \rightarrow C$ for some B and C . In this case, $A^t = C^t$. If (a) holds, then $C \sim \top$ by induction hypothesis. Hence, $A \sim B \rightarrow \top \sim \top$ by $(\cong\text{-}\rightarrow\top)$. On the other hand, if (b) is the case, then $C[\top/Y] \sim \top$ by induction hypothesis. Hence, $A[\top/Y] = B[\top/Y] \rightarrow C[\top/Y] \sim B[\top/Y] \rightarrow \top \sim \top$ by $(\cong\text{-}\rightarrow\top)$.

Case: $A = \bullet B$ for some B . In this case, $A^t = \bullet B^t$. If (a) holds, then $B \sim \top$ by induction hypothesis. Therefore, $A \sim \bullet \top \sim \top$ by Proposition 4.3.1. In case (b), $B[\top/Y] \sim \top$ by induction hypothesis. Hence, $A[\top/Y] \sim \bullet B[\top/Y] \sim \bullet \top \sim \top$ by Proposition 4.3.1, again.

Case: $A = \mu Z.B$ for some Z and B . In this case, $(\mu Z.B)^t = \mu Z.B^t$, $m_0 = 0$, $Z = X_1$, $n \geq 1$ and $B^t = \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} Y$. First, suppose that (a) holds. If $i \geq 2$, then $B \sim \top$ by induction hypothesis. Hence, $B[\top/Z] \sim \top[\top/Z] = \top$ by Proposition 4.4; and therefore, $\mu Z.B \sim \top$ by $(\cong\text{-uniq})$. On the other hand, if $i = 1$, i.e., $Y = Z$, then $B[\top/Z] \sim \top$ by induction hypothesis, since $Z = Y \notin \{X_2, X_3, \dots, X_n\}$. Therefore, again $A = \mu Z.B \sim \top$ by $(\cong\text{-uniq})$. Second, suppose that (b) holds. We get $B[\top/Y] \sim \top$ by induction hypothesis. Hence, $A[\top/Y] = (\mu Z.B)[\top/Y] = \mu Z.B[\top/Y] \sim \mu Z.\top \sim \top[\mu Z.\top/Z] = \top$ by Proposition 4.12.1 and $(\cong\text{-fix})$. \square

Theorem 4.14. A type expression A is a \top -variant if and only if $A \sim \top$.

Proof. The “only if” part is straightforward from Lemma 4.13.1. For the “if” part, suppose that $A \sim \top$, and let $A^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} Y$. We get $A^t \sim \top^t = \top$ by Proposition 4.9 and Definition 2.2. Hence, $Y = X_i$ for some i ($1 \leq i \leq n$) by Definition 2.7 and Proposition 4.11.1, because $ETV^+(\top) = \{\}$. Therefore, taking the largest such i , we also establish $Y \notin \{X_{i+1}, X_{i+2}, \dots, X_n\}$ and $m_i + m_{i+1} + m_{i+2} + \dots + m_n \geq 1$ by Proposition 2.9.3. \square

By Theorem 4.14, we can show other basic properties about the equality of type expressions as follows. In the succeeding sections, we might use Theorem 4.14 in proofs without mention.

- Proposition 4.15.** 1. $X \not\sim \top$.
 2. $\bullet A \sim \top$ if and only if $A \sim \top$.
 3. $A \rightarrow B \sim \top$ if and only if $B \sim \top$.
 4. $A \sim \top$ if and only if $A^t \sim \top$.

Proof. Straightforward from Proposition 2.3 and Theorem 4.14. \square

Proposition 4.16. If $X \notin ETV(A)$, then $A \sim A[B/X]$.

Proof. If A is a \top -variant, then $A \sim A[B/X]$ by Proposition 2.10.1 and Theorem 4.14. For the case when A is not, by straightforward induction on $h(A)$, and by cases on the form of A . \square

4.4. Canonical forms of type expressions

So far we have confirmed that the equality defined by the derivation rules preserves the basic properties of type expressions, such as \top -variants, properness, dp_{\bullet}^{\pm} , dp_{\rightarrow}^{\pm} and ETV^{\pm} . However, we know almost nothing about when two type expressions are equal to each other. For example, one might conjecture the following properties.

1. $X \sim Y$ if and only if $X = Y$.
2. $\bullet A \sim \bullet B$ if and only if $A \sim B$.
3. $A \rightarrow B \sim C \rightarrow D$ if and only if (a) $A \sim C$ and $B \sim D$, or (b) $B \sim D \sim \top$.

Furthermore,

4. $X \not\sim \bullet A$.
5. $X \not\sim A \rightarrow B$.
6. $\bullet A \cong B \rightarrow C$ implies $A \cong C \cong \top$, which should not be the case for \cong because of $(\simeq\text{-}\mathbf{K/L})$.

In fact, Items 1, 4 and 5 can be shown by Theorems 4.7 and 4.8 considering an appropriate type environment under a certain non-trivial interpretation³. However, we need some more preparation to realize that the other expectations are also fulfilled. Because of the existence of $(\cong \rightarrow \top)$ (and also $(\simeq\text{-}\mathbf{K/L})$ in case of \simeq), the equivalence relation \cong (or \simeq) is not identical to the equality as (possibly infinite) labeled trees obtained by unfolding recursive types. So it is convenient to define canonical representation of type expressions.

Definition 4.17 (Canonical type expressions). We define a set \mathbf{CTExp}^{\cong} (respectively, \mathbf{CTExp}^{\simeq}) of *canonical* type expressions as follows.

$$\begin{aligned}\mathbf{CTExp}^{\cong} &::= \top \mid \bullet^n \mathbf{TVar} \mid \bullet^n (\mathbf{TExp}_1 \rightarrow \mathbf{TExp}_2) \\ \mathbf{CTExp}^{\simeq} &::= \top \mid \bullet^n \mathbf{TVar} \mid \mathbf{TExp}_1 \rightarrow \mathbf{TExp}_2\end{aligned}$$

where n is an arbitrary non-negative integer and \mathbf{TExp}_2 is not a \top -variant.

The following definition can be regarded as an effective procedure for finding canonical forms of given type expressions, which will be immediately verified by Proposition 4.19. Later, also by Proposition 4.21, it will be shown that $A \not\sim B$ if canonical forms of two type expressions A and B fit in different categories of the three forms above.

³Proposition 4.15.1 can be also shown in the same way.

Definition 4.18. For each type expression A , we define $A^{c\cong}$ as follows.

$$\begin{aligned}
A^{c\cong} &= \top & (A \text{ is a } \top\text{-variant}) \\
X^{c\cong} &= X \\
(\bullet A)^{c\cong} &= \bullet A^{c\cong} & (\bullet A \text{ is not a } \top\text{-variant}) \\
(A \rightarrow B)^{c\cong} &= A \rightarrow B & (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
(\mu X. A)^{c\cong} &= A^{c\cong}[\mu X. A/X] & (\mu X. A \text{ is not a } \top\text{-variant})
\end{aligned}$$

We similarly define $A^{c\sim}$ by adjusting the definition of $(\bullet A)^{c\cong}$ as follows.

$$(\bullet A)^{c\sim} = \begin{cases} \bullet \bullet^n X & (A^{c\sim} = \bullet^n X) \\ \bullet B \rightarrow \bullet C & (A^{c\sim} = B \rightarrow C) \end{cases}$$

For example, let $A = \bullet \mu X. \bullet (X \rightarrow \bullet Y)$ and $B = X \rightarrow \mu Y. X \rightarrow \bullet (Z \rightarrow Y)$. Then, $A^{c\sim} = \bullet \bullet ((\mu X. \bullet (X \rightarrow \bullet Y)) \rightarrow \bullet Y)$, $A^{c\cong} = \bullet \bullet (\mu X. \bullet (X \rightarrow \bullet Y)) \rightarrow \bullet \bullet \bullet Y$, and $B^{c\sim} = B^{c\cong} = \top$.

In order to make the description concise, we will use A^c , to denote either $A^{c\cong}$ or $A^{c\sim}$, according to the context in which we consider \cong or \sim , respectively.

Proposition 4.19. A^c is a canonical type expression such that $A^c \sim A$, that is, $A^{c\cong} \cong A$ and $A^{c\sim} \simeq A$.

Proof. If A is a \top -variant, then $A^c = \top \sim A$ by Definition 4.18 and Theorem 4.14. Hence, we assume that A is not a \top -variant; and therefore, $A \not\sim \top$. The proof proceeds by induction on $h(A)$, and by cases on the form of A . Most cases are straightforward. In the case when $A = \mu X. B$ for some X and B , we get that B^c is canonical and $B^c \sim B$ by induction hypothesis. Hence, we get $A^c = B^c[A/X] \sim B[A/X] \sim A$ by Proposition 4.4 and $(\cong\text{-fix})$, and get $B^c \neq \top$ from $A \not\sim \top$, since $B^c = \top$ implies $A = \mu X. B \sim \mu X. B^c = \mu X. \top \sim \top$ by Proposition 4.12.1 and $(\cong\text{-fix})$. Therefore,

- (a) $B^c = \bullet^n Y$ for some n and Y , or
- (b) $B^c = \bullet^n (C \rightarrow D)$ for some n , C and D such that D is not a \top -variant, where $n = 0$ in case of \simeq .

In case (a), $Y \neq X$ since $Y = X$ implies $A = \mu X. B \sim \mu X. \bullet^n X \sim \top$ by Propositions 4.12.1 and 4.3.2. Hence, A^c is canonical since $A^c = B^c[A/X] = (\bullet^n Y)[A/X] = \bullet^n Y$. As for case (b), since neither A nor D is a \top -variant, $D[A/X]$ is not a \top -variant either, by Proposition 2.10.2, which implies that $B^c[A/X]$ is canonical. Thus, A^c is a canonical type expression. \square

The following lemma will be used in Proposition 4.21.

Lemma 4.20. If neither $A[\top/X]$ nor B is a \top -variant, then $A[B/X]^c = A^c[B/X]$.

Proof. By induction on $h(A)$, and by cases on the form of A . Suppose that neither $A[\top/X]$ nor B is a \top -variant, which also implies that neither A nor $A[B/X]$ is a \top -variant by Propositions 2.10.1 and 2.10.2.

Case: $A = Y$ for some Y . In this case, we get $Y \neq X$ since $A[\top/X]$ is not a \top -variant. Therefore, $A[B/X]^c = Y[B/X]^c = Y^c = Y$ and $A^c[B/X] = Y^c[B/X] = Y[B/X] = Y$.

Case: $A = \bullet C$ for some C . Since $A[\top/X]$ is not a \top -variant, $C[\top/X]$ is not either, by Proposition 2.3.2. Hence, by induction hypothesis,

$$C[B/X]^c = C^c[B/X]. \quad (11)$$

Furthermore, C is not a \top -variant since A is not. Hence, there are two subcases on the form of C^c as follows.

- (i) $C^c = \bullet^n Y$ for some n and Y . By Definition 4.18, $A^c = \bullet^{n+1} Y$ in this case. By Propositions 4.19 and 4.4, we get $A[\top/X] \sim A^c[\top/X] = (\bullet^{n+1} Y)[\top/X]$. Thus, $Y \neq X$ since $A[\top/X]$ is not a \top -variant. Hence, $C[B/X]^c = C^c[B/X] = (\bullet^n Y)[B/X] = \bullet^n Y$ by (11). Therefore, $A[B/X]^c = (\bullet C)[B/X]^c = (\bullet C[B/X])^c = \bullet^{n+1} Y$, and $A^c[B/X] = (\bullet^{n+1} Y)[B/X] = \bullet^{n+1} Y$.

- (ii) $C^c = \bullet^n(D \rightarrow E)$ for some n , D and E , where $n = 0$ in case of \simeq . We get $A^{c\simeq} = \bullet^{n+1}(D \rightarrow E)$ and $A^{c\simeq} = \bullet D \rightarrow \bullet E$ by Definition 4.18. On the other hand, we also get $C[B/X]^c = C^c[B/X] = \bullet^n(D[B/X] \rightarrow E[B/X])$ by (11). Therefore, in case of \cong , $A[B/X]^{c\cong} = (\bullet C[B/X])^{c\cong} = \bullet^{n+1}(D[B/X] \rightarrow E[B/X])$ and $A^{c\cong}[B/X] = (\bullet^{n+1}(D \rightarrow E))[B/X] = \bullet^{n+1}(D[B/X] \rightarrow E[B/X])$. Similarly, for \simeq , $A[B/X]^{c\simeq} = (\bullet C[B/X])^{c\simeq} = \bullet D[B/X] \rightarrow \bullet E[B/X]$ and $A^{c\simeq}[B/X] = (\bullet D \rightarrow \bullet E)[B/X] = \bullet D[B/X] \rightarrow \bullet E[B/X]$.

Case: $A = C \rightarrow D$ for some C and D . Since A is not a \top -variant, D is not, either. Hence, $A^c = C \rightarrow D$ by Definition 4.18. Therefore, $A[B/X]^c = (C \rightarrow D)[B/X]^c = (C[B/X] \rightarrow D[B/X])^c = C[B/X] \rightarrow D[B/X]$, and $A^c[B/X] = (C \rightarrow D)[B/X] = C[B/X] \rightarrow D[B/X]$.

Case: $A = \mu Y.C$ for some Y and C . We assume that $Y \notin \{X\} \cup FTV(B)$ without loss of generality. Since $A[\top/X]$ is not a \top -variant, $C[\top/X]$ is not, either. Therefore, $A[B/X]^c = (\mu Y.C[B/X])^c = C[B/X]^c[\mu Y.C[B/X]/Y] = C[B/X]^c[A[B/X]/Y] = C^c[B/X][A[B/X]/Y] = C^c[B/X, A[B/X]/Y]$ by Definition 4.18 and the induction hypothesis on C . On the other hand, $A^c[B/X] = C^c[A/Y][B/X] = C^c[B/X, A[B/X]/Y]$. \square

Proposition 4.21. *Suppose that $A \sim B$.*

- (i) *If $A^c = \top$, then $B^c = \top$.*
- (ii) *If $A^c = \bullet^n X$ for some n and X , then $B^c = \bullet^n X$.*
- (iii) *If $A^c = \bullet^n(C \rightarrow D)$ for some n , C and D , then $B^c = \bullet^n(C' \rightarrow D')$ for some C' and D' such that $C \sim C'$ and $D \sim D' \not\sim \top$, where $n = 0$ in case of \simeq .*

Proof. If $A \sim B$, and if A or B is a \top -variant, then $A^c = B^c = \top$ by Definition 4.18 and Theorem 4.14. Hence, we assume that neither A nor B is a \top -variant, i.e., $A \not\sim \top$ and $B \not\sim \top$. We prove the following more general claim by induction on the derivation of $A \sim B$ or $B \sim A$.

If $A \sim B \not\sim \top$ or $B \sim A \not\sim \top$, then (ii) and (iii) hold.

Suppose that $A \sim B \not\sim \top$ or $B \sim A \not\sim \top$. By cases on the last rule applied in the derivation. Most cases are trivial. The only interesting cases are the following three.

Case: (\cong -fix). In this case, for some Y and C , $A = \mu Y.C$ and $B = C[\mu Y.C/Y]$, or vice versa. Note that $C[\top/Y] \not\sim \top$, since $C[\top/Y] \sim \top$ implies $\mu Y.C \sim \top$ by (\cong -uniq). Therefore, $(\mu Y.C)^c = C^c[\mu Y.C/Y] = C^c[\mu Y.C/Y]^c$ by Lemma 4.20.

Case: (\cong -uniq). In this case, there exist some Y , C and D such that $D \sim C[D/Y]$, and either $A = D$ and $B = \mu Y.C$, or vice versa. Similarly to the previous case, $C[\top/Y]$ is not a \top -variant; and hence, $C[D/Y]^c = C^c[D/Y]$ by Lemma 4.20. We also get $C^c \neq \top$ and $C^c \neq \bullet^n Y$ for any n , since $\mu Y.C$ is not a \top -variant. That is, there are two subcases on the form of C^c to consider.

- (a) $C^c = \bullet^n X$ for some n and X such that $X \neq Y$.
- (b) $C^c = \bullet^n(E \rightarrow F)$ for some n , E and F such that $F \not\sim \top$, where $n = 0$ in case of \simeq .

In case (a), $C[D/Y]^c = C^c[D/Y] = (\bullet^n X)[D/Y] = \bullet^n X$; and hence, $D^c = \bullet^n X$ by induction hypothesis. On the other hand, $(\mu Y.C)^c = C^c[\mu Y.C/Y] = (\bullet^n X)[\mu Y.C/Y] = \bullet^n X$. As for case (b), $C[D/Y]^c = C^c[D/Y] = (\bullet^n(E \rightarrow F))[D/Y] = \bullet^n(E[D/Y] \rightarrow F[D/Y])$; and hence, by induction hypothesis, $D^c = \bullet^n(E' \rightarrow F')$ for some E' and F' such that $E' \sim E[D/Y]$ and $F' \sim F[D/Y] \not\sim \top$. On the other hand, $(\mu Y.C)^c = C^c[\mu Y.C/Y] = (\bullet^n(E \rightarrow F))[\mu Y.C/Y] = \bullet^n(E[\mu Y.C/Y] \rightarrow F[\mu Y.C/Y])$. Note that $E' \sim E[D/Y] \sim E[\mu Y.C/Y]$ and $F' \sim F[D/Y] \sim F[\mu Y.C/Y]$ by Proposition 4.4 since $D \sim \mu Y.C$.

Case: (\simeq -K/L). This case only applies to \simeq . In this case, $A = \bullet(C \rightarrow D)$ and $B = \bullet C \rightarrow \bullet D$, or vice versa. Therefore, $A^{c\simeq} = B^{c\simeq} = \bullet C \rightarrow \bullet D$ by Definition 4.18. \square

Proposition 4.22. *$A \sim B$ if and only if either*

- (i) $A^c = B^c = \top$,

- (ii) $A^c = B^c = \bullet^n X$ for some n and X , or
- (iii) $A^c = \bullet^n(C \rightarrow D)$ and $B^c = \bullet^n(E \rightarrow F)$ for some n , C , D , E and F such that $C \sim E$ and $D \sim F$. Furthermore, $n = 0$ in case of \simeq .

Proof. The “if” part is straightforward by Proposition 4.19, $(\cong \bullet)$ and $(\cong \rightarrow)$. We get the “only if” part by Propositions 4.19 and 4.21. \square

4.5. Other properties of the equality of types

In the rest of this section, we shall continue to examine other properties of \cong and \simeq that are necessary for the proofs in the succeeding sections. The conjectures raised at the beginning of the previous subsection will be proved.

Proposition 4.23. 1. If $\bullet A \cong B \rightarrow C$ then $A \cong C \cong \top$.

- 2. If $\bullet A \simeq B \rightarrow C$, then (a) $A \simeq C \simeq \top$, or (b) $A \simeq D \rightarrow E$ for some D and E such that $\bullet D \simeq B$ and $\bullet E \simeq C$.

Proof. For Item 1, suppose that $\bullet A \cong B \rightarrow C$. Since either $A \not\cong \top$ or $C \not\cong \top$ implies that neither $\bullet A$ nor $B \rightarrow C$ is a \top -variant by Proposition 2.3 and Theorem 4.14, we get $(\bullet A)^{c\approx} = \bullet A^{c\approx}$ and $(B \rightarrow C)^{c\approx} = B \rightarrow C$ by Definition 4.18. However, it contradicts Proposition 4.22. As for Item 2, suppose that $\bullet A \simeq B \rightarrow C$, and that either $A \not\cong \top$ or $C \not\cong \top$. Since either one implies that $B \rightarrow C$ is not a \top -variant, we get $(B \rightarrow C)^{c\approx} = B \rightarrow C$. Therefore, by Propositions 4.19, 4.22 and Definition 4.18, we get $A \simeq A^{c\approx} = D \rightarrow E$ for some D and E such that $(\bullet A)^{c\approx} = \bullet D \rightarrow \bullet E$, $\bullet D \simeq B$, and $\bullet E \simeq C$. \square

Proposition 4.24. Let m and n be non-negative integers.

- 1. $\bullet^m X \not\sim \bullet^n(A \rightarrow B)$.
- 2. $\bullet^m X \not\sim \top$.
- 3. $\bullet^m X \sim \bullet^n Y$ if and only if $X = Y$ and $m = n$.
- 4. $X \not\sim \bullet A$.

Proof. Straightforward from Proposition 4.21 and Definition 4.18⁴. \square

Definition 4.25. Let n be a non-negative integer. We define $\lceil A \rceil^n$ as follows.

$$\begin{aligned}
\lceil A \rceil^n &= \top & (A \text{ is a } \top\text{-variant}) \\
\lceil X \rceil^n &= X \\
\lceil \bullet A \rceil^0 &= \bullet A & (\bullet A \text{ is not a } \top\text{-variant}) \\
\lceil \bullet A \rceil^{n+1} &= \lceil A \rceil^n & (\bullet A \text{ is not a } \top\text{-variant}) \\
\lceil A \rightarrow B \rceil^n &= \lceil A \rceil^n \rightarrow \lceil B \rceil^n & (A \rightarrow B \text{ is not a } \top\text{-variant}) \\
\lceil \mu X. A \rceil^n &= \lceil A[\mu X. A/X] \rceil^n & (\mu X. A \text{ is not a } \top\text{-variant})
\end{aligned}$$

Note that $\lceil A \rceil^n$ is defined by induction on the lexicographic ordering of $\langle n, r(A) \rangle$. For example,

$$\begin{aligned}
\lceil \mu X. \bullet X \rightarrow Y \rceil^0 &= \lceil \bullet(\mu X. \bullet X \rightarrow Y) \rceil^0 \rightarrow Y^{\top 0} \\
&= \lceil \bullet(\mu X. \bullet X \rightarrow Y) \rceil^0 \rightarrow \lceil Y \rceil^{\top 0} \\
&= \bullet(\mu X. \bullet X \rightarrow Y) \rightarrow Y,
\end{aligned}$$

⁴Proposition 4.24 can be also shown by Theorems 4.7 and 4.8 considering an appropriate type environment under a certain non-trivial interpretation.

$$\begin{aligned}
\ulcorner \mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y) \urcorner^3 &= \ulcorner \bullet (\bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \rightarrow \bullet \bullet Y) \urcorner^3 \\
&= \ulcorner \bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \rightarrow \bullet \bullet Y \urcorner^2 \\
&= \ulcorner \bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \urcorner^2 \rightarrow \ulcorner \bullet \bullet Y \urcorner^2 \\
&= \ulcorner \mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y) \urcorner^1 \rightarrow \ulcorner \bullet Y \urcorner^1 \\
&= \ulcorner \bullet (\bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \rightarrow \bullet \bullet Y) \urcorner^1 \rightarrow \ulcorner Y \urcorner^0 \\
&= \ulcorner \bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \rightarrow \bullet \bullet Y \urcorner^0 \rightarrow Y \\
&= (\ulcorner \bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \urcorner^0 \rightarrow \ulcorner \bullet \bullet Y \urcorner^0) \rightarrow Y \\
&= (\bullet (\mu X. \bullet (\bullet X \rightarrow \bullet \bullet Y)) \rightarrow \bullet \bullet Y) \rightarrow Y.
\end{aligned}$$

Proposition 4.26. 1. $\ulcorner A \urcorner^0 \sim A$.

2. $\ulcorner \bullet A \urcorner^{n+1} \sim \ulcorner A \urcorner^n$.

3. $\ulcorner A \rightarrow B \urcorner^n \sim \ulcorner A \urcorner^n \rightarrow \ulcorner B \urcorner^n$.

4. $\ulcorner \mu X. A \urcorner^n \sim \ulcorner A[\mu X. A/X] \urcorner^n$.

Proof. Item 1 is by straightforward induction on $r(A)$. Note that other items are trivial for type expressions other than \top -variants. Use Propositions 2.3, 2.10.4 and Theorem 4.14. \square

Lemma 4.27. Let m and n be non-negative integers, $A_1, \dots, A_m, B_1, \dots, B_m$ and C be type expressions such that $A_i \sim B_i$ and $\ulcorner A_i \urcorner^k \sim \ulcorner B_i \urcorner^k$ for every $k < n$ and i ($1 \leq i \leq m$). Let $\vec{X} = X_1, \dots, X_m$, $\vec{A} = A_1, \dots, A_m$ and $\vec{B} = B_1, \dots, B_m$. If for every i , either (a) C is proper in X_i , or (b) $\ulcorner A_i \urcorner^n \sim \ulcorner B_i \urcorner^n$, then $\ulcorner C[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n$.

Proof. By induction on the lexicographic ordering of $\langle n, h(C) \rangle$, and by cases on the form of C . Suppose (a) or (b) holds for every i ($1 \leq i \leq m$). If C is a \top -variant or $n = 0$, then we get $\ulcorner C[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n$ by Proposition 2.10.1, or by Propositions 4.26.1 and 4.4, respectively. Hence, we assume that C is not a \top -variant and $n > 0$.

Case: $C = Y$ for some Y . If $Y \notin \{\vec{X}\}$, then obvious since $C[\vec{A}/\vec{X}] = C = C[\vec{B}/\vec{X}]$. If $Y = X_i$ for some i , then $C[\vec{A}/\vec{X}] = A_i$ and $C[\vec{B}/\vec{X}] = B_i$. Hence, $\ulcorner C[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n$ from (b) since C is not proper in X_i .

Case: $C = \bullet D$ for some D . In this case, $\ulcorner D[\vec{A}/\vec{X}] \urcorner^{n-1} \sim \ulcorner D[\vec{B}/\vec{X}] \urcorner^{n-1}$ by induction hypothesis. Hence, $\ulcorner C[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner D[\vec{A}/\vec{X}] \urcorner^{n-1} \sim \ulcorner D[\vec{B}/\vec{X}] \urcorner^{n-1} \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n$ by Proposition 4.26.2.

Case: $C = D \rightarrow E$ for some D and E . Note that, for every i , both D and E are proper in X_i if and only if so is C , since C is not a \top -variant. Therefore, $\ulcorner D[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner D[\vec{B}/\vec{X}] \urcorner^n$ and $\ulcorner E[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner E[\vec{B}/\vec{X}] \urcorner^n$ by induction hypothesis; and hence, $\ulcorner C[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner D[\vec{A}/\vec{X}] \urcorner^n \rightarrow \ulcorner E[\vec{A}/\vec{X}] \urcorner^n \sim \ulcorner D[\vec{B}/\vec{X}] \urcorner^n \rightarrow \ulcorner E[\vec{B}/\vec{X}] \urcorner^n \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n$ by Proposition 4.26.3 and $(\cong \rightarrow)$.

Case: $C = \mu Y. D$ for some Y and D . We can assume that for any i , $Y \notin \{X_i\} \cup FTV(A_i) \cup FTV(B_i)$ without loss of generality. Note that, for every i , D is proper in X_i if and only if so is C , since C is not a \top -variant and $Y \neq X_i$. By induction hypothesis, $\ulcorner C[\vec{A}/\vec{X}] \urcorner^k \sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^k$ for every $k < n$. Since D is proper in Y ,

$$\begin{aligned}
\ulcorner C[\vec{A}/\vec{X}] \urcorner^n &\sim \ulcorner D[\vec{A}/\vec{X}][C[\vec{A}/\vec{X}]/Y] \urcorner^n && \text{(by Proposition 4.26.4)} \\
&= \ulcorner D[\vec{A}/\vec{X}, C[\vec{A}/\vec{X}]/Y] \urcorner^n && \text{(since } Y \notin FTV(\vec{A}) \text{)} \\
&\sim \ulcorner D[\vec{B}/\vec{X}, C[\vec{B}/\vec{X}]/Y] \urcorner^n && \text{(by induction hypothesis)} \\
&= \ulcorner D[\vec{B}/\vec{X}][C[\vec{B}/\vec{X}]/Y] \urcorner^n && \text{(since } Y \notin FTV(\vec{B}) \text{)} \\
&\sim \ulcorner C[\vec{B}/\vec{X}] \urcorner^n && \text{(by Proposition 4.26.4).}
\end{aligned}$$

\square

Proposition 4.28. *If $A \sim B$, then $\ulcorner A \urcorner^n \sim \ulcorner B \urcorner^n$ for every n .*

Proof. Suppose that $A \sim B$. Since obviously $\ulcorner A \urcorner^0 \sim \ulcorner B \urcorner^0$ by Proposition 4.26.1, we assume that $n > 0$, below. By induction on the derivation of $A \sim B$, and by cases on the last rule applied in the derivation. Use Proposition 4.26. The only non-trivial case is $(\cong\text{-} \text{uniq})$. In this case, there exist some X and C such that $B = \mu X.C$, $A \sim C[A/X]$ and C is proper in X . By induction hypothesis,

$$\ulcorner A \urcorner^n \sim \ulcorner C[A/X] \urcorner^n. \quad (12)$$

We show that $\ulcorner A \urcorner^n \sim \ulcorner B \urcorner^n$ by induction on n . By the induction hypothesis on n , we have

$$\ulcorner A \urcorner^k \sim \ulcorner B \urcorner^k \text{ for every } k < n. \quad (13)$$

Therefore,

$$\begin{aligned} \ulcorner A \urcorner^n &\sim \ulcorner C[A/X] \urcorner^n && \text{(by (12))} \\ &\sim \ulcorner C[B/X] \urcorner^n && \text{(by Lemma 4.27 and (13))} \\ &\sim \ulcorner B \urcorner^n && \text{(by Proposition 4.26.4).} \end{aligned} \quad \square$$

Proposition 4.29. 1. $\bullet A \sim \bullet B$ if and only if $A \sim B$.

2. $A \rightarrow B \sim C \rightarrow D$ if and only if (a) $A \sim C$ and $B \sim D$, or (b) $B \sim D \sim \top$.

Proof. The “if” part of Item 1 is obvious from $(\cong\text{-}\bullet)$, and so is the “only if” part since $A \sim \ulcorner A \urcorner^0 \sim \ulcorner \bullet A \urcorner^1 \sim \ulcorner \bullet B \urcorner^1 \sim \ulcorner B \urcorner^0 \sim B$ by Propositions 4.26.1, 4.26.2 and 4.28. The “if” part of Item 2 is also obvious from $(\cong\text{-}\rightarrow)$ and $(\cong\text{-}\rightarrow\top)$. As for the “only if” part of Item 2, suppose that $A \rightarrow B \sim C \rightarrow D$. If $A \rightarrow B$ or $C \rightarrow D$ is a \top -variant, then so are both B and D by Proposition 2.3.3 and Theorem 4.14; and hence, $B \sim D \sim \top$. On the other hand, if neither is a \top -variant, we get $A \sim C$ and $B \sim D$ by Proposition 4.21 since $(A \rightarrow B)^c = A \rightarrow B$ and $(C \rightarrow D)^c = C \rightarrow D$. \square

We shall conclude this section with the following lemma, which will be used in the proof of Propositions 5.7 and 5.8 in the succeeding section.

Lemma 4.30. *Let A be proper in X , and suppose that $A[B/X] \sim B$.*

1. *If $B^c = \bullet^n Y$, then $A^c = \bullet^n Y$.*
2. *If $B^{c\approx} = \bullet^n (C \rightarrow D)$, then $A^{c\approx} = \bullet^n (C' \rightarrow D')$ for some n , C' and D' such that $C'[B/X] \sim C$ and $D'[B/X] \sim D$, where $n = 0$ is case of \simeq .*

Proof. By cases on the form of A^c . Suppose that A is proper in X and $A[B/X] \sim B$.

Case: $A^c = \top$. Either $B^c = \bullet^n Y$ or $B^c = \bullet^n (C \rightarrow D)$ implies $B \not\sim \top$ by Proposition 4.21; and hence, $A^c \neq \top$ from $A[B/X] \sim B$ by Propositions 2.10.1, 4.19 and Theorem 4.14.

Case: $A^c = \bullet^m Z$ for some m and Z . If $Z = X$, then $m > 0$ since A is proper in X , and $B \sim A[B/X] \sim (\bullet^m X)[B/X] = \bullet^m B$, which implies $B \sim \top$ by Proposition 4.3.3. Hence, $Z \neq X$; and therefore, $B \sim A[B/X] \sim (\bullet^m Z)[B/X] = \bullet^m Z$, which implies that $B^c = \bullet^m Z$ by Proposition 4.21. Thus, both Items 1 and 2 hold.

Case: $A^c = \bullet^m (E \rightarrow F)$ for some m , E and F . In this case, $B \sim A[B/X] \sim \bullet^m (E \rightarrow F)[B/X] = \bullet^m (E[B/X] \rightarrow F[B/X])$. Hence, by Proposition 4.21, $B^c = \bullet^m (C \rightarrow D)$ for some C and D such that $C \cong E[B/X]$ and $D \cong F[B/X]$. Therefore, both Items 1 and 2 hold. Note that $m = 0$ in case of \simeq . \square

5. Subtyping

As mentioned in Section 1, our intended interpretation of the \bullet -modality introduces a subtyping relation into types. We define the subtyping relation by a set of derivation rules in a similar way to [2].

Definition 5.1. A *subtyping assumption* is a finite set of pairs of type variables such that any type variable appears at most once in the set. To denote the subtyping assumption $\{ \langle X_i, Y_i \rangle \mid i = 1, 2, \dots, n \}$, we write $\{ X_1 \preceq Y_1, X_2 \preceq Y_2, \dots, X_n \preceq Y_n \}$. We use $\gamma, \gamma', \gamma_1, \gamma_2, \dots$ to denote subtyping assumptions, and $FTV(\gamma)$ to denote the set of type variables occurring in γ .

Definition 5.2 (\preceq). We define the derivability of *subtyping judgment* $\gamma \vdash A \preceq B$ by the following derivation rules.

$$\begin{array}{c}
\frac{}{\gamma \cup \{X \preceq Y\} \vdash X \preceq Y} (\preceq\text{-assump}) \qquad \frac{}{\gamma \vdash A \preceq \top} (\preceq\text{-}\top) \\
\\
\frac{A \simeq B}{\gamma \vdash A \preceq B} (\preceq\text{-reflex}) \qquad \frac{\gamma_1 \vdash A \preceq B \quad \gamma_2 \vdash B \preceq C}{\gamma_1 \cup \gamma_2 \vdash A \preceq C} (\preceq\text{-trans}) \\
\\
\frac{\gamma \vdash A \preceq B}{\gamma \vdash \bullet A \preceq \bullet B} (\preceq\text{-}\bullet) \qquad \frac{\gamma_1 \vdash A' \preceq A \quad \gamma_2 \vdash B \preceq B'}{\gamma_1 \cup \gamma_2 \vdash A \rightarrow B \preceq A' \rightarrow B'} (\preceq\text{-}\rightarrow) \\
\\
\frac{\gamma \cup \{X \preceq Y\} \vdash A \preceq B}{\gamma \vdash \mu X. A \preceq \mu Y. B} (\preceq\text{-}\mu) \quad \left(X \notin FTV(\gamma) \cup FTV(B), Y \notin FTV(\gamma) \cup FTV(A), \right. \\
\left. \text{and } A \text{ and } B \text{ are proper in } X \text{ and } Y, \text{ respectively} \right) \\
\\
\frac{}{\gamma \vdash A \preceq \bullet A} (\preceq\text{-approx})
\end{array}$$

Note that $\gamma \cup \{X \preceq Y\}$ and $\gamma_1 \cup \gamma_2$ in the rules must be well-formed subtyping assumptions, i.e., any type variable must not have more than one occurrence in them. We also define a binary relation \preceq over \mathbf{TExp} as $A \preceq B$ if and only if $\{\} \vdash A \preceq B$ is derivable.

The subtyping relation \preceq reflects the interpretation of type expressions in $\lambda\mathbf{A}$ -frames. Most of the subtyping rules are standard. The rule $(\preceq\text{-}\mu)$ corresponds to the ‘‘Amber rule’’ [6]. The rules $(\preceq\text{-}\top)$, $(\preceq\text{-}\bullet)$ and $(\preceq\text{-approx})$ reflect our intended meaning of the \bullet -modality discussed in Section 1. In the rest of this section, we discuss some basic properties of \preceq .

We can also consider another subtyping judgment, say $\gamma \vdash A \preceq B$, by substituting $A \cong B$ for the antecedent of the $(\preceq\text{-reflex})$ -rule and by instead adding the following subtyping rule, which corresponds to the axiom schema **K** of normal modal logic.

$$\frac{}{\gamma \vdash \bullet(A \rightarrow B) \preceq \bullet A \rightarrow \bullet B} (\preceq\text{-K})$$

However, in order to concentrate on the main purpose of the present paper, we will not discuss such a variant in the sequel⁵.

Definition 5.3. Let $\langle \mathcal{W}, \triangleright \rangle$ be a $\lambda\mathbf{A}$ -frame, η a type environment, and γ a subtyping assumption. We write $\eta \models \gamma$ if and only if $\eta(X)_p \subseteq \eta(Y)_p$ for every $p \in \mathcal{W}$, X and Y such that $\{X \preceq Y\} \subseteq \gamma$.

⁵In [22], this variant is called $S\text{-}\lambda\bullet\mu^+$.

Theorem 5.4 (Soundness of \preceq). Consider the interpretation \mathcal{I} over a $\lambda\mathbf{A}$ -frame $\langle \mathcal{W}, \triangleright \rangle$. Let η be a hereditary type environment. If $\gamma \vdash A \preceq B$ and $\eta \models \gamma$, then $\mathcal{I}(A)_p^\eta \subseteq \mathcal{I}(B)_p^\eta$ for every $p \in \mathcal{W}$.

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases on the last subtyping rule applied in the derivation.

Case: (\preceq -assump). In this case, $A = X$ and $B = Y$ for some X and Y such that $\{X \preceq Y\} \subseteq \gamma$. We get $\eta(X)_p \subseteq \eta(Y)_p$ for every p from $\eta \models \gamma$.

Case: (\preceq - \top). Obvious from Definition 3.8.

Case: (\preceq -reflex). Obvious from Theorem 4.8.

Case: (\preceq -trans). For some C , γ_1 and γ_2 such that $\gamma = \gamma_1 \cup \gamma_2$, the derivation ends with

$$\frac{\gamma_1 \vdash A \preceq C \quad \gamma_2 \vdash C \preceq B}{\gamma_1 \cup \gamma_2 \vdash A \preceq B} (\preceq\text{-trans}).$$

Since $\gamma = \gamma_1 \cup \gamma_2$, we get $\eta \models \gamma_1$ and $\eta \models \gamma_2$ from $\eta \models \gamma$. Therefore, $\mathcal{I}(A)_p^\eta \subseteq \mathcal{I}(C)_p^\eta \subseteq \mathcal{I}(B)_p^\eta$ by induction hypothesis.

Case: (\preceq - \bullet). For some A' and B' such that $A = \bullet A'$ and $B = \bullet B'$, the derivation ends with

$$\frac{\gamma \vdash A' \preceq B'}{\gamma \vdash \bullet A' \preceq \bullet B'} (\preceq\text{-}\bullet).$$

We get $\mathcal{I}(A')_q^\eta \subseteq \mathcal{I}(B')_q^\eta$ for every q by induction hypothesis. Therefore, by Proposition 3.9,

$$\mathcal{I}(\bullet A')_p^\eta = \{u \mid u \in \mathcal{I}(A')_q^\eta \text{ for every } q \triangleleft p\} \subseteq \{u \mid u \in \mathcal{I}(B')_q^\eta \text{ for every } q \triangleleft p\} = \mathcal{I}(\bullet B')_p^\eta.$$

Case: (\preceq - \rightarrow). For some $A_1, A_2, B_1, B_2, \gamma_1$ and γ_2 such that $A = A_1 \rightarrow A_2$, $B = B_1 \rightarrow B_2$ and $\gamma = \gamma_1 \cup \gamma_2$, the derivation ends with

$$\frac{\gamma_1 \vdash B_1 \preceq A_1 \quad \gamma_2 \vdash A_2 \preceq B_2}{\gamma_1 \cup \gamma_2 \vdash A_1 \rightarrow A_2 \preceq B_1 \rightarrow B_2} (\preceq\text{-}\rightarrow).$$

Similarly to the previous case, we get $\mathcal{I}(B_1)_q^\eta \subseteq \mathcal{I}(A_1)_q^\eta$ and $\mathcal{I}(A_2)_q^\eta \subseteq \mathcal{I}(B_2)_q^\eta$ for every q , by induction hypothesis. Therefore, by Proposition 3.9, $\mathcal{I}(A_1 \rightarrow A_2)_p^\eta = \{u \mid u \cdot v \in \mathcal{I}(A_2)_q^\eta \text{ for every } v \in \mathcal{I}(A_1)_q^\eta \text{ whenever } p \triangleright^* q\} \subseteq \{u \mid u \cdot v \in \mathcal{I}(B_2)_q^\eta \text{ for every } v \in \mathcal{I}(B_1)_q^\eta \text{ whenever } p \triangleright^* q\} = \mathcal{I}(B_1 \rightarrow B_2)_p^\eta$.

Case: (\preceq - μ). For some X', Y', A' and B' such that $A = \mu X'.A'$ and $B = \mu Y'.B'$, the derivation ends with

$$\frac{\gamma \cup \{X' \preceq Y'\} \vdash A' \preceq B'}{\gamma \vdash \mu X'.A' \preceq \mu Y'.B'} (\preceq\text{-}\mu)$$

where $X' \notin FTV(\gamma) \cup FTV(B')$, $Y' \notin FTV(\gamma) \cup FTV(A')$, and furthermore, A' and B' are proper in X' and Y' , respectively. We show that $\mathcal{I}(A)_p^\eta \subseteq \mathcal{I}(B)_p^\eta$ for every $p \in \mathcal{W}$ by induction on p . Suppose that $p \in \mathcal{W}$. By the induction hypothesis on p ,

$$\mathcal{I}(A)_q^\eta \subseteq \mathcal{I}(B)_q^\eta \text{ for every } q \triangleleft p. \tag{14}$$

Let η' be the type environment defined as follows.

$$\begin{aligned} \eta'(X')_q &= \begin{cases} \mathcal{I}(A)_q^\eta & (p \triangleright q) \\ \{\} & (p \not\triangleright q) \end{cases} \\ \eta'(Y')_q &= \begin{cases} \mathcal{I}(B)_q^\eta & (p \triangleright q) \\ \{\} & (p \not\triangleright q) \end{cases} \\ \eta'(Z)_q &= \eta(Z)_q \quad (Z \notin \{X', Y'\}) \end{aligned}$$

Note that

$$\eta'(Z)_q = \eta[\mathcal{I}(A)^\eta/X', \mathcal{I}(B)^\eta/Y'](Z)_q \text{ for every } Z \text{ and } q \triangleleft p, \quad (15)$$

and that $\eta' \models \gamma \cup \{X' \preceq Y'\}$ by (14). Hence, by the induction hypothesis on the derivation,

$$\mathcal{I}(A')_p^{\eta'} \subseteq \mathcal{I}(B')_p^{\eta'}. \quad (16)$$

On the other hand,

$$\begin{aligned} \mathcal{I}(A)_p^\eta &= \mathcal{I}(A'[A/X'])_p^\eta && \text{(by Proposition 3.9)} \\ &= \mathcal{I}(A'[A/X', B/Y'])_p^\eta && \text{(by Proposition 4.16 and Theorem 4.8)} \\ &= \mathcal{I}(A')_p^{\eta[\mathcal{I}(A)^\eta/X', \mathcal{I}(B)^\eta/Y']} && \text{(by Proposition 3.10.1)} \\ &= \mathcal{I}(A')_p^{\eta'} && \text{(by (15) and Lemma 4.6).} \end{aligned}$$

Similarly, $\mathcal{I}(B)_p^\eta = \mathcal{I}(B')_p^{\eta'}$. Therefore, $\mathcal{I}(A)_p^\eta \subseteq \mathcal{I}(B)_p^\eta$ by (16).

Case: (\preceq -approx). $B = \bullet A$ in this case. Hence, $\mathcal{I}(A)_p^\eta \subseteq \{u \mid u \in \mathcal{I}(A)_q^\eta \text{ for every } q \triangleleft p\} = \mathcal{I}(B)_p^\eta$ by Propositions 3.10.2 and 3.9. \square

Although not being necessary for showing the soundness of subtyping judgments, the following basic propositions should also be established in order to discuss formal derivability of subtyping, later.

- Proposition 5.5.** 1. *If $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ is derivable, and $\gamma \cup \{X' \preceq Y'\}$ is a well-formed subtyping assumption, then $\gamma \cup \{X' \preceq Y'\} \vdash A[X'/X, Y'/Y] \preceq B[X'/X, Y'/Y]$ is also derivable without changing the height of derivation. (**renaming**)*
2. *If $\gamma \vdash A \preceq B$ is derivable, and γ' is a subtyping assumption such that $\gamma \subseteq \gamma'$, then $\gamma' \vdash A \preceq B$ is also derivable without changing the height of derivation. (**weakening**)*
3. *If $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ is derivable, then so is $\gamma \vdash A[C/X, C/Y] \preceq B[C/X, C/Y]$ without changing the height of derivation.*
4. *If $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ and $\gamma \vdash C \preceq D$ are derivable, then so is $\gamma \vdash A[C/X, D/Y] \preceq B[C/X, D/Y]$. (**substitution**)*

Proof. By induction on the height of the derivation of $\gamma \cup \{X \preceq Y\} \vdash A \preceq B$ or $\gamma \vdash A \preceq B$, and by cases on the last rule applied. The proofs for Items 1 and 2 proceed by simultaneous induction. We use Item 1 to rename the bound type variables in case of (\preceq - μ), and Item 2 to unify the typing assumptions of the two antecedents when the last rule is (\preceq -trans) or (\preceq - \rightarrow). Use Proposition 4.4 for the case (\preceq -reflex) in the proofs of Items 1, 3 and 4. \square

In the rest of the present section, we shall show three more basic propositions about subtyping, which give us some intuitions about the structure of subtyping introduced by the derivation rules, and are crucial to prove the subject reduction property of the typing system in the next section.

Proposition 5.6. *If $\gamma \vdash \top \preceq A$ is derivable, then $A \sim \top$.*

Proof. Suppose that $\gamma \vdash A \preceq B$ is derivable. By Theorem 4.14, it suffices to show that $A \simeq \top$ implies $B \simeq \top$. The proof proceeds by induction on the height of the derivation, and by cases on the last rule applied in the derivation. The case (\preceq -assump) is impossible by Proposition 4.24.2. The cases (\preceq - \top) and (\preceq -reflex) are trivial. The case (\preceq -trans) is straightforward by induction hypothesis. Use Propositions 4.15.2 and 4.15.3 for the cases (\preceq - \bullet), (\preceq - \rightarrow), and (\preceq -approx). If the last rule is (\preceq - μ), then the derivation ends with

$$\frac{\gamma \cup \{X \preceq Y\} \vdash A' \preceq B'}{\gamma \vdash \mu X. A' \preceq \mu Y. B'} \quad (\preceq\text{-}\mu)$$

for some X, Y, A' and B' such that $A = \mu X.A', B = \mu Y.B', X \notin FTV(B'), Y \notin FTV(A')$, and furthermore, A' and B' are proper in X and Y , respectively. By Proposition 5.5.3, we can get a derivation of $\gamma \vdash A'[\top/X] \preceq B'[\top/Y]$ from the one of $\gamma \cup \{X \preceq Y\} \vdash A' \preceq B'$ without changing the height of derivation. On the other hand, since $A \simeq A'[A/X]$ and $A \simeq \top$, we also get $A'[\top/X] \simeq \top$ by Proposition 4.4. Therefore, by induction hypothesis, $B'[\top/X] \simeq \top$; and hence, $B = \mu Y.B' \simeq \top$ by $(\simeq\text{-uniq})$. \square

Proposition 5.7. *Suppose that $\gamma \vdash A \preceq B$ be a derivable subtyping judgment, and $B \not\preceq \top$.*

1. *If $A^{c\approx} = \bullet^m X$, then $B^{c\approx} = \bullet^n Y$ for some n and Y such that $m \leq n$, and either $X = Y$ or $\{X \preceq Y\} \subseteq \gamma$.*
2. *If $B^{c\approx} = \bullet^n Y$, then $A^{c\approx} = \bullet^m X$ for some m and X such that $m \leq n$, and either $X = Y$ or $\{X \preceq Y\} \subseteq \gamma$.*

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases on the last rule applied in the derivation. Use Propositions 4.19 and 4.24.3 for the case $(\preceq\text{-assump})$, and use Propositions 4.21, 5.6 and 4.3.1 for the cases $(\preceq\text{-reflex})$, $(\preceq\text{-trans})$ and $(\preceq\text{-}\bullet)$, respectively. The case $(\preceq\text{-}\rightarrow)$ is impossible by Propositions 4.24.1 and 4.19. If the last rule is $(\preceq\text{-}\mu)$, then the derivation ends with

$$\frac{\gamma \cup \{X' \preceq Y'\} \vdash A' \preceq B'}{\gamma \vdash \mu X'.A' \preceq \mu Y'.B'} (\preceq\text{-}\mu)$$

for some X', Y', A' and B' such that $A = \mu X'.A', B = \mu Y'.B', X' \notin FTV(B'), Y' \notin FTV(A')$, and furthermore, A' and B' are proper in X' and Y' , respectively. For Item 1, suppose that $A^{c\approx} = \bullet^m X$ and $B \not\preceq \top$. Note that $X \neq X'$ by Proposition 4.19 and 4.11.1, and that $B' \not\preceq \top$ since $B' \sim \top$ implies $B \sim \top$. Since $A'[A/X'] \sim A \sim A^{c\approx} = \bullet^m X$ and A' is proper in X' , we get $A'^{c\approx} = \bullet^m X$ by Lemma 4.30.1; and therefore, by induction hypothesis, $B'^{c\approx} = \bullet^n Y$ for some $n \geq m$, and $X = Y$ or $\{X \preceq Y\} \subseteq \gamma \cup \{X' \preceq Y'\}$. We get $Y' \neq Y$ from $B \not\preceq \top$ because otherwise $B \cong \mu Y.B'^{c\approx} = \mu Y.\bullet^n Y \cong \top$ by Propositions 4.19, 4.12.1 and 4.3.2; and hence, $B^{c\approx} = B'^{c\approx}[B/Y'] = \bullet^n Y$. Note that $\{X \preceq Y\} \subseteq \gamma \cup \{X' \preceq Y'\}$ implies $\{X \preceq Y\} \subseteq \gamma$ because $X' \neq X$. Symmetrically, we can show Item 2 in this case. \square

Proposition 5.8. *Suppose that $\gamma \vdash A \preceq B$ is derivable, and $B \not\preceq \top$.*

1. *If $A^{c\approx} = C \rightarrow D$, then there exist some k, E and F such that*
 - (1a) $B^{c\approx} = E \rightarrow F$,
 - (1b) $\gamma \vdash E \preceq \bullet^k C$ and $\gamma \vdash \bullet^k D \preceq F$ are derivable.
2. *If $B^{c\approx} = E \rightarrow F$, then there exist some k, C and D such that*
 - (2a) $A^{c\approx} = C \rightarrow D$,
 - (2b) $\gamma \vdash E \preceq \bullet^k C$ and $\gamma \vdash \bullet^k D \preceq F$ are derivable.

Note that k ranges over non-negative integers.

Proof. By induction on the derivation of $\gamma \vdash A \preceq B$, and by cases on the last rule used in the derivation. Lemma 4.30.2 is crucial to the case $(\preceq\text{-}\mu)$. We will employ Theorem 4.14 and Proposition 4.19 in this proof without specific mention. First, note that $A \not\preceq \top$ from $B \not\preceq \top$ by Proposition 5.6. Hence, $A^{c\approx} = C \rightarrow D$ and $B^{c\approx} = E \rightarrow F$ imply $D \not\preceq \top$ and $F \not\preceq \top$, respectively, by Proposition 2.3.3.

Case: $(\preceq\text{-assump})$. In this case, both A and B are type variables. Hence, trivial by Proposition 4.24.1.

Case: $(\preceq\text{-}\top)$. This case is impossible since $B \not\preceq \top$.

Case: $(\preceq\text{-reflex})$. In this case, $A \simeq B$. Hence, we can get both Items 1 and 2 by Proposition 4.21.

Case: $(\preceq\text{-trans})$. In this case, $\gamma_1 \vdash A \preceq G$ and $\gamma_2 \vdash G \preceq B$ are derivable for some γ_1, γ_2 and G such that $\gamma = \gamma_1 \cup \gamma_2$. We get $G \not\preceq \top$ from $B \not\preceq \top$ by Proposition 5.6. If $A^{c\approx} = C \rightarrow D$, then by the induction hypothesis on $\gamma_1 \vdash A \preceq G$, there exist some k', E' and F' such that

$$\begin{aligned} G^{c\approx} &= E' \rightarrow F', \\ \gamma_1 \vdash E' &\preceq \bullet^{k'} C \text{ and } \gamma_1 \vdash \bullet^{k'} D \preceq F' \text{ are derivable.} \end{aligned}$$

Therefore, by the induction hypothesis on $\gamma_2 \vdash G \preceq B$, there exist some k'' , E and F such that

$$\begin{aligned} B^{c\approx} &= E \rightarrow F, \\ \gamma_2 \vdash E &\preceq \bullet^{k''} E' \text{ and } \gamma_2 \vdash \bullet^{k''} F' \preceq F \text{ are derivable.} \end{aligned}$$

Taking k as $k = k' + k''$, we get (1a) and (1b). The proof for Item 2 is just symmetrical.

Case: $(\preceq-\bullet)$. In this case, $\gamma \vdash A' \preceq B'$ is derivable for some A' and B' such that $A = \bullet A'$ and $B = \bullet B'$. We get $B' \not\vdash \top$ from $B \not\vdash \top$ by Proposition 4.3.1. If $A^{c\approx} = C \rightarrow D$, then $A'^{c\approx} = C' \rightarrow D'$ for some C' and D' such that $C = \bullet C'$ and $D = \bullet D'$ by Definition 4.18. Therefore, by induction hypothesis, there exist some k , E' and F' such that

$$\begin{aligned} B'^{c\approx} &= E' \rightarrow F', \\ \gamma \vdash E' &\preceq \bullet^k C' \text{ and } \gamma \vdash \bullet^k D' \preceq F' \text{ are derivable.} \end{aligned}$$

Hence, it suffices for Item 1 to take E and F as $E = \bullet E'$ and $F = \bullet F'$, respectively, since $B^{c\approx} = (\bullet B')^{c\approx} = \bullet E \rightarrow \bullet F$. The proof for Item 2 is just symmetrical.

Case: $(\preceq-\rightarrow)$. In this case, $\gamma_1 \vdash B_1 \preceq A_1$ and $\gamma_2 \vdash A_2 \preceq B_2$ are derivable for some $\gamma_1, \gamma_2, A_1, A_2, B_1$ and B_2 such that $\gamma = \gamma_1 \cup \gamma_2$, $A = A_1 \rightarrow A_2$ and $B = B_1 \rightarrow B_2$. If $A^{c\approx} = C \rightarrow D$, then we get $C = A_1$ and $D = A_2$ by Definition 4.18. Hence, it suffices to take k, E and F as $k = 0, E = B_1$ and $F = B_2$. Symmetrically, if $B^{c\approx} = E \rightarrow F$, we get $E = B_1$ and $F = B_2$; and hence, it suffices to take k, C and D as $k = 0, C = A_1$ and $D = A_2$.

Case: $(\preceq-\mu)$. In this case, $\gamma \cup \{X \preceq Y\} \vdash A' \preceq B'$ is derivable for some X, Y, A' and B' such that

$$A = \mu X. A', \tag{17}$$

$$B = \mu Y. B', \tag{18}$$

$$X \notin FTV(\gamma) \cup FTV(B'), \text{ and } \tag{19}$$

$$Y \notin FTV(\gamma) \cup FTV(A'). \tag{20}$$

Note that A' and B' are proper in X and Y , respectively, and that $A' \not\vdash \top$ and $B' \not\vdash \top$ from $A \not\vdash \top$ and $B \not\vdash \top$, respectively, by Definition 2.2. For Item 1, suppose that $A^{c\approx} = C \rightarrow D$. Then, $A'^{c\approx}[A/X] = A^{c\approx} = C \rightarrow D$ by Definition 4.18 and (17). Therefore, by Lemma 4.30.2, there exist some C' and D' such that

$$\begin{aligned} A'^{c\approx} &= C' \rightarrow D' \\ C &\simeq C'[A/X] \text{ and } D \simeq D'[A/X]. \end{aligned} \tag{21}$$

Hence, by induction hypothesis, there exist some k, E' and F' such that

$$(1a') \quad B'^{c\approx} = E' \rightarrow F',$$

$$(1b') \quad \gamma \cup \{X \preceq Y\} \vdash E' \preceq \bullet^k C' \text{ and } \gamma \cup \{X \preceq Y\} \vdash \bullet^k D' \preceq F' \text{ are derivable.}$$

Let E and F be as $E = E'[B/Y]$ and $F = F'[B/Y]$, respectively. Then, (1a) can be shown as follows.

$$\begin{aligned} B^{c\approx} &= B'^{c\approx}[B/Y] && \text{(by (18) and Definition 4.18)} \\ &= (E' \rightarrow F')[B/Y] && \text{(by (1a'))} \\ &= E \rightarrow F \end{aligned}$$

Note that $F \not\vdash \top$ from $B \not\vdash \top$ by Proposition 2.3.3; and hence, $X \notin ETV(E) \cup ETV(F) \cup FTV(B)$ from (19), since $ETV(E) \cup ETV(F) = ETV(E \rightarrow F) = ETV(B^{c\approx}) = ETV(B) \subseteq FTV(B) \subseteq FTV(B')$ by Definition 2.7 and Proposition 4.11.1. Therefore,

$$\begin{aligned} E &\simeq E[A/X] = E'[B/Y][A/X] = E'[A/X, B/Y], \text{ and} \\ F &\simeq F[A/X] = F'[B/Y][A/X] = F'[A/X, B/Y] \end{aligned}$$

by Proposition 4.16. On the other hand, similarly, we get $Y \notin ETV(C) \cup ETV(D) \cup FTV(A)$ from (20); and therefore, by (21),

$$\begin{aligned} C &\simeq C[B/Y] \simeq C'[A/X][B/Y] = C'[A/X, B/Y], \text{ and} \\ D &\simeq D[B/Y] \simeq D'[A/X][B/Y] = D'[A/X, B/Y]. \end{aligned}$$

Hence, (1b) can be also established, for $\gamma \vdash E'[A/X, B/Y] \preceq \bullet^k C'[A/X, B/Y]$ and $\gamma \vdash \bullet^k D'[A/X, B/Y] \preceq F'[A/X, B/Y]$ are derivable from (1b') by Proposition 5.5.4. We thus get (1a) and (1b) in this case. The proof for Item 2 is just symmetrical.

Case: (\preceq -approx). In this case, $B = \bullet A$. If $A^{c\approx} = C \rightarrow D$, then $B^{c\approx} = \bullet C \rightarrow \bullet D$ by Definition 4.18; and hence, it suffices to take k, E and F as $k = 1, E = \bullet C$ and $F = \bullet D$. Symmetrically, If $B^{c\approx} = E \rightarrow F$, then by Definition 4.18, $A^{c\approx} = C \rightarrow D$ for some C and D such that $E = \bullet C$ and $F = \bullet D$. Hence, it suffices to take k as $k = 1$. \square

It might be supposed that the following hold concerning the subtyping of type expressions.

(Wrong) If $\bullet A \preceq B$, then $B \simeq \bullet C$ for some C such that $A \preceq C$.

(Wrong) If $A \preceq \bullet B$, then either (a) $A \preceq B$, or (b) $A \simeq \bullet C$ for some C such that $C \preceq B$.

However, neither is the case. Consider the following counter-examples, respectively.

$$\begin{aligned} \bullet(X \rightarrow Y) &\simeq \bullet X \rightarrow \bullet Y \preceq X \rightarrow \bullet Y \\ (X \rightarrow \bullet Y) \rightarrow \bullet Z &\preceq (\bullet X \rightarrow \bullet Y) \rightarrow \bullet Z \simeq \bullet((X \rightarrow Y) \rightarrow Z) \end{aligned}$$

6. The typing system $\lambda\mathbf{A}$

In this section, we finally introduce a typed λ -calculus equipped with the modality and recursive types.

Definition 6.1 (Typing contexts). A *typing context* is a finite mapping that assigns a type expression to each individual variable of its domain. We use Γ, Γ', \dots to denote typing contexts, and $\{x_1 : A_1, \dots, x_m : A_m\}$ to denote a typing context whose domain is $\{x_1, \dots, x_m\}$ and that assigns A_i to x_i for every i , where A_1, \dots, A_m are type expressions, and x_1, \dots, x_m are distinct individual variables. We use $Dom(\Gamma)$ to denote the domain of Γ .

Definition 6.2 (Typing judgment). A *typing judgment* of $\lambda\mathbf{A}$ has the following form,

$$\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\} \vdash M : B,$$

where $\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\}$ is a typing context, M a λ -term, and B a type expression. We occasionally write the same judgment omitting $\{\}$ as follows.

$$x_1 : A_1, x_2 : A_2, \dots, x_n : A_n \vdash M : B$$

Note that n can be 0. We use $\bullet\Gamma$ to denote the typing context $\{x_1 : \bullet A_1, x_2 : \bullet A_2, \dots, x_n : \bullet A_n\}$ when $\Gamma = \{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\}$, and write $\Gamma' \preceq \Gamma$ if and only if $Dom(\Gamma') = Dom(\Gamma)$ and $\Gamma'(x) \preceq \Gamma(x)$ for every $x \in Dom(\Gamma)$.

We now define the typing rules of $\lambda\mathbf{A}$. According to the intended meaning of \bullet , three typing rules, namely (shift), (\top) and (\preceq) , are added to those of the simply typed λ -calculus.

Definition 6.3 (Typing rules). The typing system $\lambda\mathbf{A}$ is defined by the following derivation rules.

$$\frac{}{\Gamma \cup \{x : A\} \vdash x : A} \text{ (var)} \qquad \frac{\bullet\Gamma \vdash M : \bullet A}{\Gamma \vdash M : A} \text{ (shift)}$$

$$\begin{array}{c}
\frac{}{\Gamma \vdash M : \top} (\top) \qquad \frac{\Gamma \vdash M : A \quad A \preceq B}{\Gamma \vdash M : B} (\preceq) \\
\\
\frac{\Gamma \cup \{x : A\} \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} (\rightarrow I) \qquad \frac{\Gamma_1 \vdash M : A \rightarrow B \quad \Gamma_2 \vdash N : A}{\Gamma_1 \cup \Gamma_2 \vdash MN : B} (\rightarrow E)
\end{array}$$

The (\top) -rule can be considered almost redundant, since in fact $\Gamma \vdash M : \top$ is derivable without using (\top) for any closed λ -term M . In derivation trees of typing judgments, the rule (\preceq) is sometimes written as (\cong) or (\simeq) where it is applied for two type expressions A and B such that $A \cong B$ or $A \simeq B$, respectively. Note that two derivation rules, namely (shift) and (\preceq) , are not related to term (program) construction. These rules only reflect the relationship among types (specifications), just like the case of parametric polymorphism, and constitute a non-constructive (non-computational, or non-informative) part of logic of programming.

The readers may wonder the following rule, which corresponds to the necessitation rule of normal modal logic, is missing.

$$\frac{\Gamma_1 \vdash M : A}{\bullet \Gamma_1 \cup \Gamma_2 \vdash M : \bullet A} (\text{nec})$$

However, it will be shown that this rule is redundant to $\lambda\mathbf{A}$ as Proposition 7.4. On the other hand, the axiom schema \mathbf{K} of normal modal logic is incorporated into $\lambda\mathbf{A}$ in a stronger form as the $(\simeq\text{-}\mathbf{K}/\mathbf{L})$ -rule for type equality.

The (shift)-rule represents the fact that for every hereditary interpretation over a given well-founded frame, which is not necessarily a $\lambda\mathbf{A}$ -frame, we can extend it by adding new worlds so that every possible world p in the original frame has another world from which p is accessible. Since the interpretation of $\Gamma \vdash M : A$ in the world p is identical to the one of $\bullet \Gamma \vdash M : \bullet A$ in such a world, $\Gamma \vdash M : A$ is valid whenever so is $\bullet \Gamma \vdash M : \bullet A$. In [22], the (shift)-rule was treated as an optional rule to the core typing system, which has however turned to be sound with respect to any hereditary interpretation over well-founded frames⁶. The reader should refer to the proof of Theorem 7.3 for the details.

Example 6.4. We can derive Curry's fixed-point combinator \mathbf{Y} in $\lambda\mathbf{A}$; more precisely, the following is derivable.

$$\vdash \lambda f. (\lambda x. f(xx)) (\lambda x. f(xx)) : (\bullet X \rightarrow X) \rightarrow X$$

Let a type $A = \mu Y. \bullet Y \rightarrow X$ and a derivation Π as follows.

$$\begin{array}{c}
\Pi = \frac{\frac{\frac{}{f : \bullet X \rightarrow X \vdash f : \bullet X \rightarrow X} (\text{var}) \quad \frac{\frac{\frac{}{x : \bullet A \vdash x : \bullet A} (\text{var}) \quad \frac{}{x : \bullet A \vdash x : \bullet A} (\text{var})}{x : \bullet A \vdash x : \bullet A \rightarrow \bullet X} (\preceq)}{x : \bullet A \vdash x : \bullet A \rightarrow \bullet X} (\rightarrow E)}{x : \bullet A \vdash xx : \bullet X} (\rightarrow E)}{f : \bullet X \rightarrow X, x : \bullet A \vdash f(xx) : X} (\rightarrow I)}{f : \bullet X \rightarrow X \vdash \lambda x. f(xx) : \bullet A \rightarrow X} (\rightarrow I)
\end{array}$$

Then, we can derive \mathbf{Y} as follows.

$$\begin{array}{c}
\vdots \Pi \qquad \vdots \Pi \qquad \vdots \Pi \\
\frac{\frac{\frac{}{f : \bullet X \rightarrow X \vdash \lambda x. f(xx) : \bullet A \rightarrow X} (\rightarrow I) \quad \frac{\frac{}{f : \bullet X \rightarrow X \vdash \lambda x. f(xx) : \bullet A \rightarrow X} (\rightarrow I) \quad \frac{}{f : \bullet X \rightarrow X \vdash \lambda x. f(xx) : \bullet A \rightarrow X} (\rightarrow I)}{f : \bullet X \rightarrow X \vdash \lambda x. f(xx) : \bullet A \rightarrow X} (\rightarrow E)}{f : \bullet X \rightarrow X \vdash (\lambda x. f(xx)) (\lambda x. f(xx)) : X} (\rightarrow E)}{\vdash \lambda f. (\lambda x. f(xx)) (\lambda x. f(xx)) : (\bullet X \rightarrow X) \rightarrow X} (\rightarrow I)
\end{array}$$

⁶The (shift)-rule becomes redundant if we add intersection types to $\lambda\mathbf{A}$.

We can also observe that Turing's fixed-point combinator $(\lambda x. \lambda f. f(xxf)) (\lambda x. \lambda f. f(xxf))$ has the same type. More generally, we can derive $\vdash \mathbf{Y}_n : (\bullet X \rightarrow X) \rightarrow X$ for every n , where $\mathbf{Y}_0 = \mathbf{Y}$ and $\mathbf{Y}_{n+1} = \mathbf{Y}_n(\lambda y. \lambda f. f(yf))$.

The type $(\bullet X \rightarrow X) \rightarrow X$ gives a concise axiomatic meaning to the fixed-point combinators; it says that they can produce an element of X with a given function that works as an information pump from $\bullet X$ to X ; in other words, they provide the induction scheme discussed in Section 1. The type thus enables us to construct recursive programs using the fixed-point combinators without analyzing their computational behavior. We shall see some examples of such recursive programs in Section 9.

7. Basic properties of $\lambda\mathbf{A}$

In this section, we show the soundness of the typing system $\lambda\mathbf{A}$ with respect to the semantics of types presented in Section 3, and its subject reduction property. We begin with some basic property of the typing system.

Definition 7.1 (Skeletons and sizes of derivations). The *skeleton* of a derivation is defined to be the tree obtained from the derivation by erasing all the judgments occurring in it. That is, a skeleton is a finite tree with nodes labeled by names of the inference rules listed in Definition 6.3. The *size* of a derivation is defined to be the number of nodes involved in the skeleton of the derivation.

Proposition 7.2. *Let $\Gamma \cup \{x : A, y : A\}$ be a well-formed typing context, that is, $x \in \text{Dom}(\Gamma)$ implies $\Gamma(x) = A$, and the same for y .*

1. *If $\Gamma \cup \{x : A\} \vdash M : B$ is derivable, then so is $\Gamma \cup \{y : A\} \vdash M[y/x] : B$ without changing the skeleton of the derivation. (renaming)*
2. *If $\Gamma \vdash M : B$ is derivable, then so is $\Gamma \cup \{x : A\} \vdash M : B$ without changing the skeleton of the derivation. (weakening)*
3. *If $\Gamma \cup \{x : A\} \vdash M[x/y] : B$ is derivable, then so is $\Gamma \cup \{x : A, y : A\} \vdash M : B$ without changing the skeleton of the derivation. (separation)*
4. *Suppose that $x \notin \text{FV}(M)$. If $\Gamma \cup \{x : A\} \vdash M : B$ is derivable, then so is $\Gamma \vdash M : B$ without changing the skeleton of the derivation. (erasing)*

Proof. By induction on the skeletons of the derivations, and by cases on the rule applied last in them. The proofs of Items 1 and 2 proceed by simultaneous induction. We use Item 1 for 3, and use Item 2 for 3 and 4. Note that since we have the (\top) -rule, $\text{FV}(M) \subseteq \text{Dom}(\Gamma)$ is not always the case when $\Gamma \vdash M : B$ is derivable for some B . \square

Now we can show the soundness of $\lambda\mathbf{A}$ with respect to the semantics \mathcal{I} of type expressions over $\lambda\mathbf{A}$ -frames.

Theorem 7.3 (Soundness of $\lambda\mathbf{A}$). *If $\{x_1 : A_1, \dots, x_n : A_n\} \vdash M : B$ is derivable in $\lambda\mathbf{A}$, then for any $\lambda\mathbf{A}$ -frame $\langle \mathcal{W}, \triangleright \rangle$, $\llbracket M \rrbracket_\rho^\nu \in \mathcal{I}(B)_p^\eta$ for every $p \in \mathcal{W}$, hereditary η and ρ whenever $\rho(x_i) \in \mathcal{I}(A_i)_p^\eta$ for every i ($i = 1, 2, \dots, n$).*

Proof. By induction on the derivation, and by cases on the last rule used in it. Most cases are straightforward. Use Theorem 5.4 for the case of (\leq) . Let $\Gamma = \{x_1 : A_1, \dots, x_n : A_n\}$, and $\Gamma \vdash M : B$ be a derivable judgment. Suppose that $\langle \mathcal{W}, \triangleright \rangle$ is a $\lambda\mathbf{A}$ -frame, and that $\rho(x_i) \in \mathcal{I}(A_i)_p^\eta$ ($i = 1, 2, \dots, n$).

Case: (var). In this case, $M = x_j$ and $B = A_j$ for some j ($1 \leq j \leq n$). Therefore, by assumption, $\llbracket M \rrbracket_\rho^\nu = \llbracket x_j \rrbracket_\rho^\nu = \rho(x_j) \in \mathcal{I}(A_j)_p^\eta = \mathcal{I}(B)_p^\eta$.

Case: (shift). In this case, the derivation ends with

$$\frac{\{x_1 : \bullet A_1, x_2 : \bullet A_2, \dots, x_n : \bullet A_n\} \vdash M : \bullet B}{\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\} \vdash M : B} \text{ (shift)}.$$

For the given $\langle \mathcal{W}, \triangleright \rangle$, η and p , we construct another $\lambda\mathbf{A}$ -frame $\langle \mathcal{W}', \triangleright' \rangle$ and a hereditary type environment η' for $\langle \mathcal{W}', \triangleright' \rangle$ by extending $\langle \mathcal{W}, \triangleright \rangle$ and η , respectively, as follows.

$$\begin{aligned}\mathcal{W}' &= \{\star\} \cup \mathcal{W} & (\star \notin \mathcal{W}) \\ q \triangleright' r &\text{ iff } \begin{cases} q = \star \text{ and } r = p, \text{ or} \\ q, r \in \mathcal{W} \text{ and } q \triangleright r \end{cases} \\ \eta'(X)_q &= \begin{cases} \eta(X)_p & (q = \star) \\ \eta(X)_q & (q \in \mathcal{W}) \end{cases}\end{aligned}$$

where \star is a fresh world added to the original ones. Obviously, \triangleright' is (conversely) well-founded, and η' hereditary. Observe also that \triangleright' is locally linear; and hence, $\langle \mathcal{W}', \triangleright' \rangle$ constitutes a $\lambda\mathbf{A}$ -frame. Then, let $\mathcal{I}'(A)^{\eta'}$ be the interpretation of A in the $\lambda\mathbf{A}$ -frame $\langle \mathcal{W}', \triangleright' \rangle$ under the type environment η' . Note that $\mathcal{I}'(A)^{\eta'}_q = \mathcal{I}(A)^{\eta}_q$ for every $q \in \mathcal{W}$ and A . Since $\rho(x_i) \in \mathcal{I}(A_i)^{\eta}_p$, we get $\rho(x_i) \in \mathcal{I}'(A_i)^{\eta'}_q$ for every $q \triangleleft' \star$ by the definition of \triangleright' . Hence, $\rho(x_i) \in \mathcal{I}'(\bullet A_i)^{\eta'}_{\star}$. By induction hypothesis, we get $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}'(\bullet B)^{\eta'}_{\star}$; and hence, $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}'(B)^{\eta'}_p = \mathcal{I}(B)^{\eta}_p$ because $\star \triangleright' p$.

Case: (\top) . Obvious since $\mathcal{I}(\top)^{\eta}_p = \mathcal{V}$ for every $p \in \mathcal{W}$ by Definition 3.8.

Case: (\preceq) . For some B' , the derivation ends with

$$\frac{\Gamma \vdash M : B' \quad \vdash B' \preceq B}{\Gamma \vdash M : B} (\preceq).$$

We get $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}(B)^{\eta}_p$ by Theorem 5.4 because $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}(B')^{\eta}_p$ by induction hypothesis.

Case: $(\rightarrow\text{I})$. The derivation ends with

$$\frac{\Gamma \cup \{y : B_1\} \vdash L : B_2}{\Gamma \vdash \lambda y. L : B_1 \rightarrow B_2} (\rightarrow\text{I})$$

for some y , L , B_1 and B_2 such that $M = \lambda y. L$ and $B = B_1 \rightarrow B_2$. Suppose that $p \triangleleft^* r$ and $v \in \mathcal{I}(B_1)^{\eta}_r$. By Proposition 3.9, it suffices to show that $\llbracket \lambda y. L \rrbracket_{\rho}^{\mathcal{V}} \cdot v \in \mathcal{I}(B_2)^{\eta}_r$. Since $p \triangleleft^* r$, by Proposition 3.10.2, $\rho[v/y](x_i) = \rho(x_i) \in \mathcal{I}(A_i)^{\eta}_p \subseteq \mathcal{I}(A_i)^{\eta}_r$ for every i such that $x_i \neq y$, and $\rho[v/y](y) = v \in \mathcal{I}(B_1)^{\eta}_p \subseteq \mathcal{I}(B_1)^{\eta}_r$. Hence, by induction hypothesis, $\llbracket L \rrbracket_{\rho[v/y]}^{\mathcal{V}} \in \mathcal{I}(B_2)^{\eta}_r$. Therefore, $\llbracket \lambda y. L \rrbracket_{\rho}^{\mathcal{V}} \cdot v = \llbracket L \rrbracket_{\rho[v/y]}^{\mathcal{V}} \in \mathcal{I}(B_2)^{\eta}_r$ by Definition 3.4.

Case: $(\rightarrow\text{E})$. The derivation ends with

$$\frac{\Gamma_1 \vdash M_1 : C \rightarrow B \quad \Gamma_2 \vdash M_2 : C}{\Gamma_1 \cup \Gamma_2 \vdash M_1 M_2 : B} (\rightarrow\text{E})$$

for some Γ_1 , Γ_2 , M_1 , M_2 and C such that $\Gamma = \Gamma_1 \cup \Gamma_2$ and $M = M_1 M_2$. Since $\Gamma_1, \Gamma_2 \subseteq \Gamma$, by induction hypothesis, $\llbracket M_1 \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}(C \rightarrow B)^{\eta}_p$ and $\llbracket M_2 \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}(C)^{\eta}_p$. Hence, $\llbracket M_1 M_2 \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M_1 \rrbracket_{\rho}^{\mathcal{V}} \cdot \llbracket M_2 \rrbracket_{\rho}^{\mathcal{V}} \in \mathcal{I}(B)^{\eta}_p$ by Definition 3.4 and Proposition 3.9. \square

Theorem 7.3 assures us that the modularity of programs is preserved even if we regard type expressions, or specifications, as asserting the convergence of programs. For example, if a type B comprises values that are normalizable to canonical ones, and we have a program M of a type $A \rightarrow B$, then we can expect that M terminates and returns such a canonical value when we provide a value of A . In Section 8, we will show such convergence properties of well-typed λ -terms by a discussion on the soundness with respect to a term model of the untyped λ -calculus.

In the rest of the present section, we will show other important properties of the typing system, in particular, the subject reduction property of the system, which is also necessary to derive a certain convergence

property of well-typed λ -terms (cf. Theorem 8.34). First, we show that the following two rules are derivable in $\lambda\mathbf{A}$. This makes the proof of the subject reduction much easier.

$$\frac{\Gamma_1 \vdash M : A}{\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A} \text{ (nec)} \quad \frac{\Gamma_1 \cup \{x : A\} \vdash M : B \quad \Gamma_2 \vdash N : A}{\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : B} \text{ (subst)}$$

Proposition 7.4. *The two rules (nec) and (subst) are derivable in $\lambda\mathbf{A}$. That is, the following two hold.*

1. *Let $\bullet\Gamma_1 \cup \Gamma_2$ be a well-formed typing context. If $\Gamma_1 \vdash M : A$ is derivable, then so is $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$.*
2. *Let $\Gamma_1 \cup \Gamma_2$ be a well-formed typing context. If $\Gamma_1 \cup \{x : A\} \vdash M : B$ and $\Gamma_2 \vdash N : A$ are derivable, then so is $\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : B$.*

Proof. Each proof proceeds by induction on the size of the derivation for M , and each induction step is shown by cases on the last rule applied in it. For Item 1, suppose that $\Gamma_1 \vdash M : A$ is derivable.

Case: (var). In this case, M is an individual variable and $\{M : A\} \subseteq \Gamma_1$. Therefore, $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$ is derivable by (var).

Case: (shift). The derivation ends with

$$\frac{\bullet\Gamma_1 \vdash M : \bullet A}{\Gamma_1 \vdash M : A} \text{ (shift)}.$$

Since $\bullet\bullet\Gamma_1 \cup \bullet\Gamma_2 \vdash M : \bullet\bullet A$ is derivable by induction hypothesis, so is $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$ by (shift).

Case: (\top). In this case, $A = \top$. Hence, $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$ is derivable by (\top) and (\preceq) since $\bullet A = \bullet\top \simeq \top$ by Proposition 4.3.1.

Case: (\preceq). Straightforward by induction hypothesis since $A' \preceq A$ implies $\bullet A' \preceq \bullet A$ for every A' by ($\preceq\bullet$).

Case: (\rightarrow I). The derivation ends with

$$\frac{\Gamma_1 \cup \{x : B\} \vdash K : C}{\Gamma_1 \vdash \lambda x. K : B \rightarrow C} (\rightarrow\text{I})$$

for some B, C, x and K such that $A = B \rightarrow C$ and $M = \lambda x. K$. By induction hypothesis, $\bullet\Gamma_1 \cup \{x : \bullet B\} \cup \Gamma_2 \vdash K : \bullet C$ is derivable. Hence, so is $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$ by (\rightarrow I) and (\preceq) since $\bullet B \rightarrow \bullet C \simeq \bullet(B \rightarrow C)$. Note that the proof would fail at this point if we did not have the ($\simeq\mathbf{K/L}$)-rule, and even if we added an extra subtyping rule $\bullet(B \rightarrow C) \preceq \bullet B \rightarrow \bullet C$ corresponding to the axiom schema \mathbf{K} of normal modal logic.

Case: (\rightarrow E). In this case, the derivation ends with

$$\frac{\Gamma_{11} \vdash M_1 : B \rightarrow A \quad \Gamma_{12} \vdash M_2 : B}{\Gamma_{11} \cup \Gamma_{12} \vdash M_1 M_2 : A} (\rightarrow\text{E})$$

for some $\Gamma_{11}, \Gamma_{12}, B, M_1$ and M_2 such that $\Gamma_1 = \Gamma_{11} \cup \Gamma_{12}$ and $M = M_1 M_2$. By induction hypothesis, $\bullet\Gamma_{11} \cup \Gamma_2 \vdash M_1 : \bullet(B \rightarrow A)$ and $\bullet\Gamma_{12} \cup \Gamma_2 \vdash M_2 : \bullet B$ are derivable. Hence, so is $\bullet\Gamma_1 \cup \Gamma_2 \vdash M : \bullet A$ by (\preceq) and (\rightarrow E) since $\bullet(B \rightarrow A) \preceq \bullet B \rightarrow \bullet A$. This completes the proof of Item 1.

The proof for Item 2 is almost straightforward. The only non-trivial case is when the last rule is (shift). In this case, the derivation ends with

$$\frac{\bullet\Gamma_1 \cup \{x : \bullet A\} \vdash M : \bullet B}{\Gamma_1 \cup \{x : A\} \vdash M : B} \text{ (shift)}.$$

Since $\Gamma_2 \vdash N : A$ is derivable, so is $\bullet\Gamma_2 \vdash N : \bullet A$ by Item 1 of this proposition. Therefore, $\bullet\Gamma \vdash M[N/x] : \bullet B$ is also derivable by induction hypothesis; and hence, so is $\Gamma \vdash M[N/x] : B$ by (shift). Note that the (nec)-rule, or Item 1 of this proposition, is crucial to this case. \square

Note that Proposition 7.4 depends on the $(\simeq\mathbf{K}/\mathbf{L})$ -rule. Without this equality, even with the normal subtyping rule $\bullet(A \rightarrow B) \preceq \bullet A \rightarrow \bullet B$, and even if we dropped the (shift)-rule from $\lambda\mathbf{A}$, the (nec)-rule would not be derivable in $\lambda\mathbf{A}$. Furthermore, without the $(\simeq\mathbf{K}/\mathbf{L})$ -rule, even if we added the (nec)-rule to $\lambda\mathbf{A}$, the (subst)-rule would remain underivable, because we could not handle the case of the (nec)-rule in the proof of Proposition 7.4.2. For example, we have the following two derivations in such a variant of $\lambda\mathbf{A}$.

$$\frac{\frac{\frac{g : X \rightarrow Z \vdash g : X \rightarrow Z \quad x : X \vdash x : X}{g : X \rightarrow Z, x : X \vdash gx : Z} (\rightarrow E)}{x : X \vdash \lambda g. gx : (X \rightarrow Z) \rightarrow Z} (\rightarrow I)}{x : \bullet X \vdash \lambda g. gx : \bullet((X \rightarrow Z) \rightarrow Z)} (\text{nec}) \quad \frac{f : Y \rightarrow \bullet X \vdash f : Y \rightarrow \bullet X \quad y : Y \vdash y : Y}{f : Y \rightarrow \bullet X, y : Y \vdash fy : \bullet X} (\rightarrow E)$$

However, we cannot derive the following judgment without the (subst)-rule⁷.

$$f : Y \rightarrow \bullet X, y : Y \vdash \lambda g. g(fy) : \bullet((X \rightarrow Z) \rightarrow Z)$$

Our final task in this section is to prove the subject reduction property of $\lambda\mathbf{A}$. To this end, we prepare the following three lemmas.

Lemma 7.5. *Suppose that $\Gamma' \preceq \Gamma$. If $\Gamma \vdash M : B$ is derivable, then so is $\Gamma' \vdash M : B$.*

Proof. Suppose that $\Gamma \cup \{x : A\} \vdash M : B$ is derivable, $A' \preceq A$, and $x \notin \text{Dom}(\Gamma)$. It suffices to show that $\Gamma \cup \{x : A'\} \vdash M : B$ is also derivable. Let y be a fresh individual variable. Since $\{y : A'\} \vdash y : A$ is derivable by (var) and (\preceq) , so is $\Gamma \cup \{y : A'\} \vdash M[y/x] : B$ by Proposition 7.4.2. Therefore, $\Gamma \cup \{x : A'\} \vdash M : B$ is also derivable by Proposition 7.2.1. \square

By this proposition, we realize that the following extension of the (\preceq) rule does not affect the derivability of typing judgments⁸.

$$\frac{\Gamma \vdash M : A \quad \Gamma' \preceq \Gamma \quad A \preceq A'}{\Gamma' \vdash M : A'} (\preceq)$$

Lemma 7.6. *Suppose that $A \rightarrow B \preceq \bullet^n(C \rightarrow D) \not\preceq \top$. Then $\bullet^n C \preceq \bullet^k A$ and $\bullet^k B \preceq \bullet^n D$ for some k .*

Proof. Note that neither $A \rightarrow B$ nor $C \rightarrow D$ is a \top -variant by Propositions 2.3.2, 5.6 and Theorem 4.14. Hence, $(A \rightarrow B)^{c\approx} = A^{c\approx} \rightarrow B^{c\approx}$ and $(\bullet^n(C \rightarrow D))^{c\approx} = \bullet^n C^{c\approx} \rightarrow \bullet^n D^{c\approx}$. Therefore, $\bullet^n C^{c\approx} \preceq \bullet^k A^{c\approx}$ and $\bullet^k B^{c\approx} \preceq \bullet^n D^{c\approx}$ for some k by Proposition 5.8, that is, $\bullet^n C \preceq \bullet^k A$ and $\bullet^k B \preceq \bullet^n D$ by Proposition 4.19. \square

Lemma 7.7 (Anti-abstraction lemma). *Let $\Gamma \vdash \lambda x. M : A$ be a derivable judgment, and suppose that $x \notin \text{Dom}(\Gamma)$. Then, $\bullet^n \Gamma \cup \{x : B\} \vdash M : C$ is also derivable for some n , B and C such that $B \rightarrow C \preceq \bullet^n A$.*

Proof. If $A \simeq \top$, then $\Gamma \cup \{x : \top\} \vdash M : \top$ is derivable by (\top) . Therefore, we assume that $A \not\preceq \top$. The proof proceeds by induction on the derivation, and by cases on the last rule applied in it. Note that neither (var) nor $(\rightarrow E)$ is possible because of the form of the λ -term $\lambda x. M$.

Case: (shift). This is the only case that we have to consider n not being 0. In this case, the derivation ends with

$$\frac{\bullet \Gamma \vdash \lambda x. M : \bullet A}{\Gamma \vdash \lambda x. M : A} (\text{shift}).$$

⁷In [22], although the author conjectured that two typing systems $S\text{-}\lambda\bullet\mu$ and $F\text{-}\lambda\bullet\mu$ described in the paper enjoy some basic properties (Proposition 2), such as the substitution lemma and the subject reduction property, it turned to be wrong. However, fortunately, this does not affect other results, including the main ones, presented in the paper.

⁸The typing system presented in [22] was formulated with this variant.

Note that $\bullet A \not\vdash \top$ from $A \not\vdash \top$. By induction hypothesis, $\bullet^{n'} \bullet \Gamma \cup \{x : B\} \vdash M : C$ is derivable for some n' , B and C such that $B \rightarrow C \preceq \bullet^{n'} \bullet A$. Therefore, it suffices to take n as $n = n' + 1$.

Case: (\top) . This case is impossible by the assumption that $A \not\vdash \top$.

Case: (\preceq) . In this case, the derivation ends with

$$\frac{\Gamma \vdash \lambda x. M : A' \quad A' \preceq A}{\Gamma \vdash \lambda x. M : A} (\preceq)$$

for some A' . By induction hypothesis, $\bullet^n \Gamma \cup \{x : B\} \vdash M : C$ is derivable for some n , B and C such that $B \rightarrow C \preceq \bullet^n A'$, which implies $B \rightarrow C \preceq \bullet^n A$ since $A' \preceq A$.

Case: $(\rightarrow I)$. Trivial since the derivation ends with

$$\frac{\Gamma \cup \{x : B\} \vdash M : C}{\Gamma \vdash \lambda x. M : B \rightarrow C} (\rightarrow I)$$

for some B and C such that $A = B \rightarrow C$. □

Finally, we can proceed to the proof of the subject reduction property of $\lambda \mathbf{A}$ with three lemmas above.

Theorem 7.8 (Subject reduction). *Suppose that $\Gamma \vdash M : A$ is derivable, and that $M \xrightarrow{\beta} L$. Then, $\Gamma \vdash L : A$ is also derivable.*

Proof. By induction on the derivation of $\Gamma \vdash M : A$, and by cases on the last rule applied in it.

Case: (var) . This case is impossible since $M \xrightarrow{\beta} L$.

Case: (\top) . In this case, $A = \top$. Hence, $\Gamma \vdash L : A$ is also derivable by (\top) .

Case: (shift) or (\preceq) . The derivation ends with either of the following (for some A' in case of (\preceq)).

$$\frac{\bullet \Gamma \vdash M : \bullet A}{\Gamma \vdash M : A} (\text{shift}) \quad \frac{\Gamma \vdash M : A' \quad A' \preceq A}{\Gamma \vdash M : A} (\preceq)$$

Hence, straightforward from the induction hypothesis by applying the same rule.

Case: $(\rightarrow I)$. The derivation ends with

$$\frac{\Gamma \cup \{x : B\} \vdash M' : C}{\Gamma \vdash \lambda x. M' : B \rightarrow C} (\rightarrow I)$$

for some x , M' , B and C such that $M = \lambda x. M'$ and $A = B \rightarrow C$. Since $\lambda x. M' \xrightarrow{\beta} L$, there exists some L' such that $M' \xrightarrow{\beta} L'$ and $L = \lambda x. L'$. Hence, by induction hypothesis, $\Gamma \cup \{x : B\} \vdash L' : C$ is derivable; and therefore, so is $\Gamma \vdash \lambda x. L' : B \rightarrow C$ by applying $(\rightarrow I)$.

Case: $(\rightarrow E)$. The derivation ends with

$$\frac{\Gamma_1 \vdash M_1 : B \rightarrow A \quad \Gamma_2 \vdash M_2 : B}{\Gamma_1 \cup \Gamma_2 \vdash M_1 M_2 : A} (\rightarrow E)$$

for some M_1 , M_2 , Γ_1 , Γ_2 and B such that $M = M_1 M_2$ and $\Gamma = \Gamma_1 \cup \Gamma_2$. If $A \simeq \top$, then $\Gamma \vdash L : A$ is derivable by (\top) . Therefore, we assume that $A \not\vdash \top$ in the sequel. Note that it also implies that $B \rightarrow A \not\vdash \top$. Since $M_1 M_2 \xrightarrow{\beta} L$, there are three possible subcases as follows:

- (a) $M_1 \xrightarrow{\beta} L_1$ and $L = L_1 M_2$ for some L_1 .
- (b) $M_2 \xrightarrow{\beta} L_2$ and $L = M_1 L_2$ for some L_2 .

(c) $M_1 = \lambda x. L'$ and $L = L'[M_2/x]$ for some x and L' .

By induction hypothesis, $\Gamma_1 \vdash L_1 : B \rightarrow A$ and $\Gamma_2 \vdash L_2 : B$ are derivable for (a) and (b), respectively. Therefore, we can derive $\Gamma \vdash L : A$ by applying (\rightarrow E) in these cases. As for (c), we can assume that $x \notin \text{Dom}(\Gamma_1)$ without loss of generality. Hence, by Lemma 7.7, there exist some n , C and D such that

$$C \rightarrow D \preceq \bullet^n(B \rightarrow A), \text{ and} \quad (22)$$

$$\bullet^n \Gamma_1 \cup \{x : C\} \vdash L' : D \text{ is derivable.} \quad (23)$$

Therefore, by (22) and Lemma 7.6,

$$\bullet^n B \preceq \bullet^k C \text{ and } \bullet^k D \preceq \bullet^n A \quad (24)$$

for some k . We can derive $\bullet^{n+k} \Gamma_1 \cup \{x : \bullet^k C\} \vdash L' : \bullet^k D$ from (23) by applying Proposition 7.4.1, namely (nec), k times; and hence, $\bullet^n \Gamma_1 \cup \{x : \bullet^k C\} \vdash L' : \bullet^k D$ is also derivable by Lemma 7.5 since $\bullet^n \Gamma_1 \preceq \bullet^{n+k} \Gamma_1$. On the other hand, $\bullet^n \Gamma_2 \vdash M_2 : \bullet^n B$ is derivable from $\Gamma_2 \vdash M_2 : B$ by (nec); and hence, so is $\bullet^n \Gamma_2 \vdash M_2 : \bullet^k C$ by (\preceq) and (24). Therefore, $\bullet^n \Gamma \vdash L'[M_2/x] : \bullet^k D$ is derivable by Proposition 7.4.2, namely (subst); and hence, so is $\bullet^n \Gamma \vdash L'[M_2/x] : \bullet^n A$ by (\preceq) and (24), from which $\Gamma \vdash L'[M_2/x] : A$ is derivable by applying (shift) n times. \square

The subject reduction property is independent of the existence of the (shift)-rule. Even if (shift) is dropped from $\lambda\mathbf{A}$, Proposition 7.4 still holds, and the statements and the proofs of Lemmas 7.6, 7.7 and Theorem 7.8 remain valid by regarding n as 0.

8. Convergence of well-typed λ -terms

When we gave the interpretation of type expressions in Definition 3.8, the only restriction forced on what type variables mean was that the interpretation must be hereditary with respect to the accessibility relation. Therefore, just as in the case of the simply typed lambda calculus, we are free to incorporate convergence properties of inhabitants into the meaning of type expressions even though recursive types are involved. Thus, type expressions can still say something about convergence, and the soundness theorem assures the convergence of well-typed λ -terms according to their types. In this section, we give such results as follows.

1. Every λ -term of types other than \top is head normalizable, i.e., solvable.
2. Every λ -term of types without positive occurrences of \top has a Böhm tree without unsolvable terms.
3. Every λ -term of types without occurrences of the modal operator \bullet is normalizable.

8.1. A term model

The results about the convergence of well-typed λ -terms are proved by applying the soundness of $\lambda\mathbf{A}$ (Theorem 7.3) to an appropriate interpretation of type expressions, considering a certain term model of the untyped λ -calculus. So we first introduce the term model used for that purpose.

Proposition 8.1. *If $M \equiv_{\beta} M'$ and $N \equiv_{\beta} N'$, then $M[N/x] \equiv_{\beta} M'[N'/x]$.*

Proof. Show that $N \equiv_{\beta} N'$ implies $M[N/x] \equiv_{\beta} M[N'/x]$, and $M \rightarrow_{\beta} M'$ implies $M[N/x] \equiv_{\beta} M'[N/x]$ by induction on the structure of M . \square

Definition 8.2. The term model $\langle \mathcal{V}, \cdot, \llbracket \cdot \rrbracket^{\mathcal{V}} \rangle$ of \mathbf{Exp} is defined as follows.

1. $\mathcal{V} = \mathbf{Exp} / \equiv_{\beta}$
2. $\llbracket M \rrbracket \cdot \llbracket N \rrbracket = \llbracket MN \rrbracket$
3. $\llbracket M \rrbracket_{\rho}^{\mathcal{V}} = \llbracket M[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k] \rrbracket$

where $[M]$ and $\tilde{\rho}(z)$ denote the equivalence class including M and the representative member of $\rho(z)$, respectively, and z_1, z_2, \dots, z_k are distinct individual variables such that $FV(M) = \{z_1, z_2, \dots, z_k\}$. Note that Proposition 8.1 assures the well-definedness of Item 3.

Proposition 8.3. *The term model $\langle \mathcal{V}, \cdot, \llbracket \cdot \rrbracket^\mathcal{V} \rangle$ is a syntactical λ -algebra.*

Proof. It suffices to show that the eight conditions of Definition 3.4 are satisfied by the term model. Conditions 1 through 5 are straightforward. As for Condition 6, let $x \notin FV(\tilde{\rho}(z_i)) \cup \{z_i\}$ for every i ($i = 1, 2, \dots, k$).

$$\begin{aligned}
\llbracket \lambda x. M \rrbracket_\rho^\mathcal{V} \cdot [N] &= [(\lambda x. M)[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k]] \cdot [N] && \text{(by Definition 8.2)} \\
&= [\lambda x. (M[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k])] \cdot [N] \\
&\quad \text{(since } x \notin FV(\tilde{\rho}(z_i)) \cup \{z_i\} \text{ for every } i) \\
&= [(\lambda x. (M[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k])) N] && \text{(by Definition 8.2)} \\
&= [M[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k][N/x]] \\
&= [M[\tilde{\rho}(z_1)/z_1, \tilde{\rho}(z_2)/z_2, \dots, \tilde{\rho}(z_k)/z_k, N/x]] \\
&\quad \text{(since } x \notin FV(\tilde{\rho}(z_i)) \cup \{z_i\} \text{ for every } i) \\
&= \llbracket M \rrbracket_{\rho[N/x]}^\mathcal{V} && \text{(by Definition 8.2)}
\end{aligned}$$

Conditions 7 and 8 are also straightforward from Definition 8.2 and Proposition 8.1. \square

8.2. Tail finite types

In this subsection, we show that every λ -term of a certain class of type expressions, which will later be defined as *tail finite* types, is *head normalizable*. The set of head normalizable terms is defined in the standard manner as follows.

Definition 8.4 (Head normal forms). A λ -term M is a *head normal form* if and only if M has the form of $\lambda x_1. \lambda x_2. \dots \lambda x_m. y N_1 N_2 \dots N_n$, where $m, n \geq 0$. We say that M *has a head normal form*, or *is head normalizable*, if $M \xrightarrow[\beta]{*} M'$ for some head normal form M' .

We also define Böhm trees of λ -terms in the standard manner according to this definition of head normal forms, in which λ -terms without head normal forms are denoted by \perp .

Definition 8.5 (Tail finite types). A type expression A is *tail finite* if and only if $A \cong \bullet^{m_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m_0, m_1, m_2, \dots, m_n, B_1, B_2, \dots, B_n$ and X .

While we used \cong to define the tail finiteness, it is also possible to use \simeq instead of \cong . Later, by Proposition 8.10, we shall see that either of the two definitions leads us to the identical notion of tail finiteness. Actually, it will be shown that a type expression A is tail finite if and only if A is not a \top -variant. To this end, we need an alternative definition of tail finiteness as follows.

Definition 8.6. Let V be a set of type variables. We define a subset \mathbf{TF}^V of \mathbf{TExp} as follows.

$$\begin{aligned}
\mathbf{TF}^V &::= X && (X \in \mathbf{TVar} - V) \\
&| \bullet \mathbf{TF}^V \\
&| \mathbf{TExp} \rightarrow \mathbf{TF}^V \\
&| \mu X. A && (\mu X. A \in \mathbf{TExp} \text{ and } A \in \mathbf{TF}^{V \cup \{X\}}).
\end{aligned}$$

We can easily check that \mathbf{TF}^V is closed under α -conversion of type expressions. We denote $\mathbf{TF}^{\{\}} by \mathbf{TF} , which will be shown to be identical to the set of tail finite type expressions. Roughly, \mathbf{TF}^V is the set of type expressions that are tail finite even when some of type variables listed in V are instantiated by \top , which will also be shown in Proposition 8.10.$

Proposition 8.7. *Let V be a set of type variables, and suppose that $A \in \mathbf{TF}^V$. Then $A \sim \bullet^{m_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m_0, m_1, m_2, \dots, m_n, B_1, B_2, B_3, \dots, B_n$ and $X \notin V$.*

Proof. By induction on $h(A)$, and by cases on the form of A .

Case: $A = Y$ for some Y . Obvious since we get $Y \notin V$ by Definition 8.6.

Case: $A = \bullet C$ for some C . We get $C \in \mathbf{TF}^V$ by Definition 8.6. Hence, by induction hypothesis, $C \sim \bullet^{m'_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m'_0, m_1, m_2, \dots, m_n, B_1, B_2, \dots, B_n$ and $X \notin V$. Therefore, it suffices to take $m_0 = m'_0 + 1$.

Case: $A = C \rightarrow D$ for some C and D . We get $D \in \mathbf{TF}^V$ by Definition 8.6. By induction hypothesis, $D \sim \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \bullet^{m_3}(B_4 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m_1, m_2, \dots, m_n, B_2, B_3, \dots, B_n$ and $X \notin V$. Therefore, it suffices to take $m_0 = 0$ and $B_1 = C$.

Case: $A = \mu Z.C$ for some Z and C . We get $C \in \mathbf{TF}^{V \cup \{Z\}}$ by Definition 8.6. Hence, by induction hypothesis, $C \sim \bullet^{m_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m_0, m_1, m_2, \dots, m_n, B_1, B_2, \dots, B_n$ and $X \notin V \cup \{Z\}$. Therefore, $A = \mu Z.C \sim C[A/Z] \sim (\bullet^{m_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots))))[A/Z] = \bullet^{m_0}(B_1[A/Z] \rightarrow \bullet^{m_1}(B_2[A/Z] \rightarrow \bullet^{m_2}(B_3[A/Z] \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n[A/Z] \rightarrow \bullet^{m_n} X) \dots)))$ by Proposition 4.4. \square

It is also not difficult to see that \mathbf{TF} enjoys the following properties.

Proposition 8.8. 1. *If $V \subseteq V'$, then $\mathbf{TF}^{V'} \subseteq \mathbf{TF}^V$.*

2. *If $X \notin \text{ETV}^+(A)$ and $A \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V \cup \{X\}}$.*

Proof. By induction on $h(A)$, and by cases on the form of A . For Item 2, note that $A \in \mathbf{TF}^V$ implies $\text{ETV}(A) \neq \{\}$ by Propositions 8.7 and 4.11.1; and hence, A is not a \top -variant, since $\text{ETV}(B) = \{\}$ for every \top -variant B by Proposition 2.9.5. \square

Proposition 8.9. *Let A be a type expression, and V a set of type variables.*

1. *If $A \in \mathbf{TF}^{V \cup \{X\}}$, then $A[B/X] \in \mathbf{TF}^V$.*
2. *If $A \in \mathbf{TF}^V$ and $B \in \mathbf{TF}^V$, then $A[B/X] \in \mathbf{TF}^V$.*
3. *If $A[B/X] \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V - \{X\}}$.*
4. *If $A[B/X] \in \mathbf{TF}^V$, then $A \in \mathbf{TF}^{V \cup \{X\}}$ or $B \in \mathbf{TF}^V$.*

Proof. By induction on $h(A)$, and by cases on the form of A . The only non-trivial case is when $A = Y$ for some Y . For Item 1, we get $X \neq Y$ from $A \in \mathbf{TF}^{V \cup \{X\}}$. Therefore, $A[B/X] = A \in \mathbf{TF}^{V \cup \{X\}} \subseteq \mathbf{TF}^V$ by Proposition 8.8.1. For Item 2, suppose that $A \in \mathbf{TF}^V$ and $B \in \mathbf{TF}^V$. We similarly get $A[B/X] = A \in \mathbf{TF}^V$ if $X \neq Y$. Otherwise, i.e., if $X = Y$, then $A[B/X] = B \in \mathbf{TF}^V$. For Items 3 and 4, suppose that $A[B/X] \in \mathbf{TF}^V$. If $X \neq Y$, then $A = A[B/X] \in \mathbf{TF}^V$; therefore, $A \in \mathbf{TF}^{V - \{X\}}$ by Proposition 8.8.1, and $A \in \mathbf{TF}^{V \cup \{X\}}$ by Proposition 8.8.2, since $X \notin \text{FTV}(A)$. Otherwise, i.e., if $X = Y$, then $A = X \in \mathbf{TF}^{V - \{X\}}$ by Definition 8.6, and $B = A[B/X] \in \mathbf{TF}^V$. \square

Now we can establish the equivalence of the two definitions. A type expression A is tail finite if and only if $A \in \mathbf{TF}$. This is shown by proving the following proposition.

Proposition 8.10. *Let V be a set of type variables. The following three conditions are equivalent.*

- (a) $A \in \mathbf{TF}^V$
- (b) $A \sim \bullet^{m_0}(B_1 \rightarrow \bullet^{m_1}(B_2 \rightarrow \bullet^{m_2}(B_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B_n \rightarrow \bullet^{m_n} X) \dots)))$ for some $n, m_0, m_1, m_2, \dots, m_n, B_1, B_2, B_3, \dots, B_n$ and $X \notin V$.
- (c) $A[\top/Y_1, \dots, \top/Y_k] \not\sim \top$, where Y_1, \dots, Y_k are distinct type variables such that $V = \{Y_1, \dots, Y_k\}$.

Note that in case of \simeq , Condition (b) is equivalent to

(b') $A \simeq B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_n \rightarrow \bullet^m X$ for some $n, m, B_1, B_2, B_3, \dots, B_n$ and $X \notin V$.

Proof. It suffices to show the following three propositions: (a) \Rightarrow (b), (b) \Rightarrow (c), and (c) \Rightarrow (a). The first one has been already shown as Proposition 8.7. The rule $(\cong \rightarrow \top)$ of Definition 4.1 is crucial to prove the last part. However, we can even show (b) \Rightarrow (a) independently of the rule.

Proof of “(b) \Rightarrow (c)”. Suppose that (b) holds, and let $B'_i = B_i[\top/Y_1, \dots, \top/Y_k]$ ($i = 1, 2, 3, \dots, n$). Since $X \notin \{Y_1, \dots, Y_k\}$, we get $A[\top/Y_1, \dots, \top/Y_k] \sim \bullet^{m_0}(B'_1 \rightarrow \bullet^{m_1}(B'_2 \rightarrow \bullet^{m_2}(B'_3 \rightarrow \dots \rightarrow \bullet^{m_{n-1}}(B'_n \rightarrow \bullet^{m_n} X) \dots)))$ by Proposition 4.4, which is not a \top -variant. Therefore, $A[\top/Y_1, \dots, \top/Y_k] \not\sim \top$ by Theorem 4.14.

Proof of “(c) \Rightarrow (a)”. We prove the contrapositive by induction on $h(A)$, and by cases on the form of A . Suppose that (a) is not the case, i.e., $A \notin \mathbf{TF}^V$. Let Y_1, \dots, Y_k be distinct type variables such that $V = \{Y_1, \dots, Y_k\}$. It suffices to show that $A[\top/Y_1, \dots, \top/Y_k] \sim \top$.

Case: $A = Y$ for some Y . We get $Y \in V$ from $A \notin \mathbf{TF}^V$ by Definition 8.6. Therefore, $A[\top/Y_1, \dots, \top/Y_k] = \top$.

Case: $A = \bullet C$ for some C . By Definition 8.6, $C \notin \mathbf{TF}^V$. By induction hypothesis, $C[\top/Y_1, \dots, \top/Y_k] \sim \top$. Hence, $A[\top/Y_1, \dots, \top/Y_k] = \bullet C[\top/Y_1, \dots, \top/Y_k] \sim \bullet \top \sim \top$ by Proposition 4.3.1.

Case: $A = C \rightarrow D$ for some C and D . Similarly, $D \notin \mathbf{TF}^V$ by Definition 8.6. Hence, by induction hypothesis, $D[\top/Y_1, \dots, \top/Y_k] \sim \top$. Therefore, $A[\top/Y_1, \dots, \top/Y_k] = C[\top/Y_1, \dots, \top/Y_k] \rightarrow D[\top/Y_1, \dots, \top/Y_k] \sim C[\top/Y_1, \dots, \top/Y_k] \rightarrow \top \sim \top$ by $(\cong \rightarrow \top)$ of Definition 4.1.

Case: $A = \mu Z.C$ for some Z and C . By Definition 8.6, $C \notin \mathbf{TF}^{V \cup \{Z\}}$. By induction hypothesis, $C[\top/Z, \top/Y_1, \dots, \top/Y_k] \sim \top$. Therefore, $A[\top/Y_1, \dots, \top/Y_k] = \mu Z.C[\top/Y_1, \dots, \top/Y_k] \sim \top$ by $(\cong\text{-uniq})$ of Definition 4.1. \square

Theorem 8.11. *The following four conditions are equivalent.*

- (a) $A \in \mathbf{TF}$
- (b) A is tail finite.
- (c) $A \not\sim \top$.
- (d) A is not a \top -variant.

Proof. Obvious from Theorem 4.14 and Proposition 8.10 considering the case that $V = \{\}$. This theorem fails if we adopt F-semantics of types by dropping $(\cong \rightarrow \top)$ from Definition 4.2. However, (a) \Leftrightarrow (b) remains to hold even in such a case. \square

Finally, we show that every λ -term of tail finite types is head normalizable, by applying the soundness of $\lambda\mathbf{A}$ with respect to the term model. The proof is quite parallel to the standard method using convertibility predicates due to Tait[27].

Definition 8.12. We define a subset \mathcal{K}_h of $\mathbf{Exp}/\overline{\beta}$ as follows.

$$\mathcal{K}_h = \{ [xN_1N_2\dots N_n] \mid x \in \mathbf{Var}, n \geq 0 \text{ and } N_i \in \mathbf{Exp} \text{ for every } i (i = 1, 2, \dots, n) \}.$$

Lemma 8.13. *Consider the term model of \mathbf{Exp} , a well-founded frame $\langle \mathcal{W}, \triangleright \rangle$ and a type environment η such that $\mathcal{K}_h \subseteq \eta(X)_p$ for every type variable X and $p \in \mathcal{W}$. Then, $\mathcal{K}_h \subseteq \mathcal{I}(A)_p^\eta$ for every A and $p \in \mathcal{W}$.*

Proof. By induction on the lexicographic ordering of $\langle p, r(A) \rangle$, and by cases on the form of A . We assume that A is not a \top -variant, since $\mathcal{K}_h \subseteq \mathcal{I}(A)_p^\eta = \mathcal{V}$ if A is. For this proof, η need not to be hereditary.

Case: $A = Y$ for some Y . By Definition 3.8, $\mathcal{I}(A)_p^\eta = \eta(Y)_p \supseteq \mathcal{K}_h$.

Case: $A = \bullet B$ for some B . In this case, $\mathcal{I}(A)_p^\eta = \{ u \mid u \in \mathcal{I}(B)_q^\eta \text{ for every } q \triangleleft p \} \supseteq \mathcal{K}_h$ since $\mathcal{K}_h \subseteq \mathcal{I}(B)_q^\eta$ for every $q \triangleleft p$ by induction hypothesis.

Case: $A = B \rightarrow C$ for some B and C . By Definition 3.8, $\mathcal{I}(A)_p^\eta = \{ u \mid u \cdot v \in \mathcal{I}(C)_q^\eta \text{ for every } v \in \mathcal{I}(B)_q^\eta \text{ whenever } p \dot{\succ}^* q \}$. Suppose that $[xN_1N_2 \dots N_n] \in \mathcal{K}_h$. Then, for every $L \in \mathbf{Exp}$, $[xN_1N_2 \dots N_n] \cdot [L] = [xN_1N_2 \dots N_nL] \in \mathcal{K}_h$. Hence, for every q such that $p \dot{\succ}^* q$, we get $[xN_1N_2 \dots N_nL] \in \mathcal{I}(C)_q^\eta$ since $\mathcal{K}_h \subseteq \mathcal{I}(C)_q^\eta$ by induction hypothesis. Therefore, $[xN_1N_2 \dots N_n] \in \mathcal{I}(B \rightarrow C)_p^\eta$.

Case: $A = \mu Y.B$ for some Y and B . By Definition 3.8, $\mathcal{I}(A)_p^\eta = \mathcal{I}(B[A/Y])_p^\eta$. On the other hand, since $r(B[A/Y]) < r(A)$ by Proposition 2.13.2, we get $\mathcal{K}_h \subseteq \mathcal{I}(B[A/Y])_p^\eta$ by induction hypothesis. \square

Note that since every $\lambda\mathbf{A}$ -frame is also a well-founded frame, Lemma 8.13 also holds in the case that $\langle \mathcal{W}, \triangleright \rangle$ is a $\lambda\mathbf{A}$ -frame. Now we show the main results of this subsection, which is stated as the following theorem.

Theorem 8.14. *Let V be a set of type variables, and A a type expression such that $A \in \mathbf{TF}^V$. If $\Gamma \vdash M : A$ is derivable in $\lambda\mathbf{A}$, then M has a head normal form.*

Proof. Consider the term model and a $\lambda\mathbf{A}$ -frame $\langle \mathbb{N}, > \rangle$ where \mathbb{N} is the set of natural numbers and $>$ is the greater-than relation between natural numbers. Note that $\langle \mathbb{N}, > \rangle$ is also a well-founded frame. Let ρ be an individual environment such that $\rho(x) = [x]$ for every $x \in \mathbf{Var}$. Then, $\rho(x) \in \mathcal{K}_h \subseteq \mathcal{I}(\Gamma(x))_p^\eta$ for every $x \in \text{Dom}(\Gamma)$, η and $p \in \mathbb{N}$ by Lemma 8.13. Hence, $[M] \in \mathcal{I}(A)_p^\eta$ for every η and p by Theorem 7.3 because $\llbracket M \rrbracket_\rho^\eta = [M]$. Since η can be any hereditary type environment, it suffices to show that M has a head normal form whenever

- (a) $A \in \mathbf{TF}^V$,
- (b) $\mathcal{K}_h \subseteq \eta(X)_p$ for every $p \in \mathbb{N}$ and X ,
- (c) $\eta(X)_p = \mathcal{K}_h$ for every $p \in \mathbb{N}$ and $X \notin V$, and
- (d) $[M] \in \mathcal{I}(A)_p^\eta$ for every $p \in \mathbb{N}$.

The proof proceeds by induction on $h(A)$, and by cases on the form of A . Suppose (a) through (d).

Case: $A = X$ for some X . In this case, $X \notin V$ from (a); and hence, $[M] \in \mathcal{I}(A)_p^\eta = \eta(X)_p = \mathcal{K}_h$ by (c) and (d). Therefore, M has a head normal form.

Case: $A = \bullet B$ for some B . In this case, $h(B) < h(A)$ and $B \in \mathbf{TF}^V$ by (a). Therefore, M has a head normal form by induction hypothesis, since $[M] \in \mathcal{I}(\bullet B)_{p+1}^\eta = \mathcal{I}(B)_p^\eta$ for every p .

Case: $A = B \rightarrow C$ for some B and C . In this case, $h(C) < h(A)$ and $C \in \mathbf{TF}^V$ by (a). Let y be a fresh individual variable. Since $[M] \in \mathcal{I}(B \rightarrow C)_p^\eta$ and $[y] \in \mathcal{K}_h \subseteq \mathcal{I}(B)_p^\eta$ for every p by (b) and Lemma 8.13, we get $[My] \in \mathcal{I}(C)_p^\eta$ for every p . Therefore, My has a head normal form, say L , by induction hypothesis. There are two possible cases: for some K , (i) $M \xrightarrow[\beta]{*} K$ and $L = Ky$, or (ii) $M \xrightarrow[\beta]{*} \lambda y. K$ and $K \xrightarrow[\beta]{*} L$. In either case, M obviously has a head normal form.

Case: $A = \mu Y.B$ for some Y and B . In this case, $h(B) < h(A)$ and $B \in \mathbf{TF}^{V \cup \{Y\}}$ by (a). By Definition 3.8 and Proposition 3.10.1, we get $\mathcal{I}(\mu Y.B)_p^\eta = \mathcal{I}(B[\mu Y.B/Y])_p^\eta = \mathcal{I}(B)_p^{\eta'}$, where $\eta' = \eta[\mathcal{I}(\mu Y.B)^\eta/Y]$. Note that (a') $B \in \mathbf{TF}^{V \cup \{Y\}}$, (b') $\mathcal{K}_h \subseteq \eta'(X)_p$ for every p and X since $\mathcal{K}_h \subseteq \mathcal{I}(\mu Y.B)^\eta = \eta'(Y)$ by (b) and Lemma 8.13, (c') $\eta'(X)_p = \mathcal{K}_h$ for every p and $X \notin V \cup \{Y\}$, and (d') $[M] \in \mathcal{I}(B)_p^{\eta'}$ for every p . Therefore, M has a head normal form by induction hypothesis. \square

Corollary 8.15. *Every λ -term of tail finite types is head normalizable.*

Proof. By Theorems 8.11 and 8.14 taking V as $V = \{\}$ in 8.14. \square

8.3. Positively and negatively finite types

In the previous subsection, it has been shown that every tail finite type, i.e., any type other than \top -variants, is inhabited only by head normalizable λ -terms. It can be also shown that, if the type does not involve any positive (effective) occurrence of \top -variants, then the Böhm tree of any λ -term that belongs to the type does not include any occurrence of \perp . For example, we can see that Curry's fixed-point combinator Y has such a Böhm tree since it has a type $(\bullet X \rightarrow X) \rightarrow X$.

Definition 8.16 (Maximal λ -terms). A λ -term M is *maximal* if and only if the Böhm tree of M has no occurrence of \perp , i.e., λ -terms that are not head normalizable.

Note that the maximality of λ -terms is closed under \equiv_{β} .

Definition 8.17 (Positively/negatively finite types). A type expression A is *positively* (respectively, *negatively*) *finite* if and only if C is tail finite whenever $A \sim B[C/X]$ for some B and X such that $X \in ETV^+(B)$ and $X \notin ETV^-(B)$ (respectively, $X \in ETV^-(B)$ and $X \notin ETV^+(B)$).

Although this definition might seem to be dependent on the choice of \sim , either of the two ways of definition gives the same notion, and the following discussion in this subsection does not depend on which it is. That will be verified by showing that the equivalent notion can be defined by the same grammar of type expressions (cf. Definition 8.18 and Proposition 8.29).

Obviously, every positively finite type expression is tail finite, and positively (negatively) finiteness is closed under \sim . In this subsection, we show that every λ -term of positively finite types is maximal. To this end, we again employ alternative definitions of positively and negatively finiteness as follows.

Definition 8.18. We define subsets **PF** and **NF** of **TExp** as follows.

$$\begin{aligned}
\mathbf{PF} &::= \mathbf{TVar} \\
&| \bullet \mathbf{PF} \\
&| \mathbf{NF} \rightarrow \mathbf{PF} \\
&| \mu X. A \quad (\mu X. A \in \mathbf{TF}, A \in \mathbf{PF}, \text{ and (a) } X \notin ETV^-(A) \text{ or (b) } A \in \mathbf{NF}) \\
\mathbf{NF} &::= \mathbf{TVar} \\
&| \bullet \mathbf{NF} \\
&| \mathbf{PF} \rightarrow \mathbf{NF} \\
&| \mu X. A \quad (\mu X. A \in \mathbf{TExp}, A \in \mathbf{NF}, \text{ and (a) } X \notin ETV^-(A) \text{ or (b) } \mu X. A \in \mathbf{PF}) \\
&| A \quad (A \text{ is a } \top\text{-variant})
\end{aligned}$$

We can easily check that **PF** and **NF** are closed under α -conversion of type expressions. Obviously, every finite type expression, i.e., one that involves no recursive type, belongs to both **PF** and **NF**. The grammar of **NF** is ambiguous since it says that any \top -variant belongs to **NF**. Hence, we should be careful with the fact that $A \rightarrow B \in \mathbf{NF}$ does not always imply $A \in \mathbf{PF}$.

Our first task in this subsection is to verify that A is positively (respectively, negatively) finite if and only if $A \in \mathbf{PF}$ (respectively, **NF**) (cf. Proposition 8.29), which also implies that it is decidable whether a given type expression is positively (negatively) finite or not.

Proposition 8.19. $\mathbf{PF} \subseteq \mathbf{TF}$.

Proof. Prove that $A \in \mathbf{PF}$ implies $A \in \mathbf{TF}$, by straightforward induction on $h(A)$, and by cases on the form of A . \square

The following proposition says that we can separate an effective occurrence of an individual variable in a type expression from others so that it becomes either positive or negative and not both. For example, consider a type expression $A = \mu Y. \bullet Y \rightarrow X$, in which the occurrence of X is both positive and negative. We can obtain a positive and not negative occurrence of X by unfolding A to $\bullet(\mu Y. \bullet Y \rightarrow X) \rightarrow X$, in which so is the rightmost occurrence of X . Similarly, we can also obtain a negative and not positive occurrence of X by unfolding A to $\bullet(\bullet(\mu Y. \bullet Y \rightarrow X) \rightarrow X) \rightarrow X$.

Proposition 8.20. *If $X \in ETV^\pm(A)$, then there exist some A' and X' such that $A \sim A'[X/X']$, $X' \in ETV^\pm(A')$ and $X' \notin ETV^\mp(A')$.*

Proof. By Propositions 4.19 and 4.11.1, we can assume that A is canonical without loss of generality. Note that $X \in ETV^\pm(A)$ implies that $A \neq \top$ and $dp_{\rightarrow}^\pm(A, X) < \infty$ by Propositions 2.9.5 and 2.15.1, respectively. By induction on $dp_{\rightarrow}^\pm(A, X)$, and by cases on the form of A . \square

Proposition 8.21. *Suppose that $B \not\sim \top$. If $A[B/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}), then $A \in \mathbf{PF}$ (respectively, \mathbf{NF}).*

Proof. If $A[B/X]$ is a \top -variant, then $A[B/X] \notin \mathbf{PF}$ by Proposition 8.19 and Theorem 8.11, and $A \in \mathbf{NF}$ since A is also a \top -variant by Proposition 2.10.2. Hence, it suffices to only consider the case that $A[B/X]$ is not a \top -variant. By induction on $h(A)$, and by cases on the form of A . The only interesting case is when $A = \mu Y. C$ for some Y and C . In this case, we can assume that $Y \notin FTV(B) \cup \{X\}$ without loss of generality. Since $A[B/X] = \mu Y. C[B/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}),

$$\mu Y. C[B/X] \in \mathbf{TF} \text{ (respectively, } \mathbf{TExp}) \quad (25)$$

$$C[B/X] \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}), \text{ and} \quad (26)$$

$$C[B/X] \in \mathbf{NF} \text{ (respectively, } \mu Y. C[B/X] \in \mathbf{PF}), \text{ if } Y \in ETV^-(C[B/X]). \quad (27)$$

First, in case of $A[B/X] \in \mathbf{PF}$, we get $\mu Y. C \in \mathbf{TF}$ from (25) by Theorem 8.11 and Proposition 2.10.1. Second, $C \in \mathbf{PF}$ (respectively, \mathbf{NF}) from (26) by induction hypothesis. Finally, if $Y \in ETV^-(C)$, then $Y \in ETV^-(C[B/X])$ by Proposition 2.11; and therefore, $C[B/X] \in \mathbf{NF}$ (respectively, \mathbf{PF}) from (27) and Definition 8.18, which implies $C \in \mathbf{NF}$ (respectively, \mathbf{PF}) by induction hypothesis. Thus, $A \in \mathbf{PF}$ (respectively, \mathbf{NF}). \square

Lemma 8.22. *Suppose that*

- (a) $A \in \mathbf{TF}^V$,
- (b) $V \cap ETV^+(B) = \{\}$, and
- (c) $X \notin ETV^+(A)$ or $B \in \mathbf{TF}$.

Then, $A[B/X] \in \mathbf{TF}^V$.

Proof. If $X \notin ETV^+(A)$, then $A \in \mathbf{TF}^{V \cup \{X\}}$ by (a) and Proposition 8.8.2; and therefore, $A[B/X] \in \mathbf{TF}^V$ by Proposition 8.9.1. On the other hand, if $X \in ETV^+(A)$, then $B \in \mathbf{TF}$ by (c); that is, $B \in \mathbf{TF}^V$ by (b) and Proposition 8.8.2. Hence, $A[B/X] \in \mathbf{TF}^V$ by (a) and Proposition 8.9.2. \square

Proposition 8.23. *Suppose that*

- (a) $A \in \mathbf{PF}$ (respectively, \mathbf{NF}),
- (b) if $X \in ETV^+(A)$ then $B \in \mathbf{PF}$ (respectively, \mathbf{NF}), and
- (c) if $X \in ETV^-(A)$ then $B \in \mathbf{NF}$ (respectively, \mathbf{PF}).

Then $A[B/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}).

Proof. By induction on $h(A)$, and by cases on the form of A . Use Proposition 2.10.1 if $A = C \rightarrow D$ for some C and D . The only interesting case is when $A = \mu Y.C$ for some Y and C . In this case, suppose that (a) through (c) hold. We can assume that $Y \notin FTV(B) \cup \{X\}$ without loss of generality; that is, $A[B/X] = \mu Y.C[B/X]$ and B is proper in Y by Proposition 2.15.6. By (a) and Definition 8.18, we have

$$\mu Y.C \in \mathbf{TF} \text{ (respectively, } \mathbf{TExp}) \quad (28)$$

$$C \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}), \text{ and} \quad (29)$$

$$Y \notin ETV^-(C) \text{ or } C \in \mathbf{NF} \text{ (respectively, } \mu Y.C \in \mathbf{PF}). \quad (30)$$

Moreover, since $ETV^\pm(C) - \{Y\} \subseteq ETV^\pm(A)$, from (b) and (c),

(b') if $X \in ETV^+(C)$ then $B \in \mathbf{PF}$ (respectively, \mathbf{NF}), and

(c') if $X \in ETV^-(C)$ then $B \in \mathbf{NF}$ (respectively, \mathbf{PF}).

By Definition 8.18, it suffices to show that

$$\mu Y.C[B/X] \in \mathbf{TF} \text{ (respectively, } \mathbf{TExp}) \quad (31)$$

$$C[B/X] \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}), \text{ and} \quad (32)$$

$$Y \notin ETV^-(C[B/X]) \text{ or } C[B/X] \in \mathbf{NF} \text{ (respectively, } \mu Y.C[B/X] \in \mathbf{PF}). \quad (33)$$

First, $C[B/X]$ is proper in Y by Proposition 2.13.3, since so is C by (28). Furthermore, in case of $A \in \mathbf{PF}$, we also get $C \in \mathbf{TF}^{\{Y\}}$ from (28); and hence, $C[B/X] \in \mathbf{TF}^{\{Y\}}$ by Lemma 8.22, Proposition 8.19 and (b'). Thus, we get (31). Second, we get (32) from (29), (b') and (c') by induction hypothesis. Finally, to show (33), suppose that $Y \in ETV^-(C[B/X])$. Then, $Y \in ETV^-(C)$ by Proposition 2.11.2 since $Y \notin FTV(B)$; and therefore, $ETV^\pm(A) = ETV(C) - \{Y\}$. Hence, from (b) and (c), we get

$$X \in ETV(C) \text{ implies } B \in \mathbf{PF} \cap \mathbf{NF}. \quad (34)$$

On the other hand, from $Y \in ETV^-(C)$ and (30), we get $C \in \mathbf{NF}$ (respectively, $\mu Y.C \in \mathbf{PF}$; and therefore, $C \in \mathbf{PF}$ by Definition 8.18). Hence, $C[B/X] \in \mathbf{NF}$ (respectively, \mathbf{PF}) from (34) by induction hypothesis, where in case of $A \in \mathbf{NF}$, $C[B/X] \in \mathbf{PF}$ also implies $\mu Y.C[B/X] \in \mathbf{PF}$ because $C[B/X] \in \mathbf{NF}$ is already established as (32), and because we can get $\mu Y.C[B/X] \in \mathbf{TF}$ from $\mu Y.C \in \mathbf{PF} \subseteq \mathbf{TF}$ by Lemma 8.22, Proposition 8.19 and (34). We thus establish (33). \square

Proposition 8.24. *Suppose that $B \not\vdash \top$.*

1. *If $X \in ETV^+(A)$ and $A[B/X] \in \mathbf{PF}$, then $B \in \mathbf{PF}$.*
2. *If $X \in ETV^-(A)$ and $A[B/X] \in \mathbf{NF}$, then $B \in \mathbf{PF}$.*

Proof. By simultaneous induction on $h(A)$, and by cases on the form of A . Suppose that $X \in ETV^+(A)$ and $A[B/X] \in \mathbf{PF}$ (respectively, $X \in ETV^-(A)$ and $A[B/X] \in \mathbf{NF}$). Note that $A \not\vdash \top$ since $X \in ETV(A)$; and hence, $A[B/X] \not\vdash \top$ from $B \not\vdash \top$ by Proposition 2.10.2 and Theorem 4.14.

Case: $A = Y$ for some Y . In this case, the assumption $X \in ETV^-(A)$ of Item 2 does not hold. As for Item 1, $Y = X$ since $X \in ETV^+(A)$. Hence, $B = A[B/X] \in \mathbf{PF}$.

Case: $A = \bullet C$ for some C . In this case, $A[B/X] = \bullet C[B/X]$. Hence, $X \in ETV^+(C)$ and $C[B/X] \in \mathbf{PF}$ (respectively, $X \in ETV^-(C)$ and $C[B/X] \in \mathbf{NF}$). Therefore, $B \in \mathbf{PF}$ by induction hypothesis.

Case: $A = C \rightarrow D$ for some C and D . Since $A[B/X] = C[B/X] \rightarrow D[B/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}), we get $X \in ETV^-(C) \cup ETV^+(D)$, $C[B/X] \in \mathbf{NF}$ and $D[B/X] \in \mathbf{PF}$ (respectively, $X \in ETV^+(C) \cup ETV^-(D)$, $C[B/X] \in \mathbf{PF}$ and $D[B/X] \in \mathbf{NF}$). Therefore, $B \in \mathbf{PF}$ by induction hypothesis.

Case: $A = \mu Y.C$ for some Y and C . We can assume that $Y \notin FTV(B) \cup \{X\}$ without loss of generality. Since $A[B/X] = \mu Y.C[B/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}), we get

$$C[B/X] \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}), \text{ and} \quad (35)$$

$$C[B/X] \in \mathbf{NF} \text{ (respectively, } \mu Y.C[B/X] \in \mathbf{PF}), \text{ if } Y \in ETV^-(C[B/X]). \quad (36)$$

On the other hand, since $X \in ETV^+(A)$ (respectively, $ETV^-(A)$), either

$$X \in ETV^+(C) \text{ (respectively, } ETV^-(C)\text{), or} \quad (37)$$

$$Y \in ETV^-(C) \text{ and } X \in ETV^-(C) \text{ (respectively, } ETV^+(C)\text{).} \quad (38)$$

Note that $Y \in ETV^-(C)$ implies $Y \in ETV^-(C[B/X])$ by Proposition 2.11, and that $\mu Y.C[B/X] \in \mathbf{PF}$ implies $C[B/X] \in \mathbf{PF}$. Therefore, from (35), (36), (37) and (38), we get either

$$\begin{aligned} &X \in ETV^+(C) \text{ (respectively, } ETV^-(C)\text{) and } C[B/X] \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}\text{), or} \\ &X \in ETV^-(C) \text{ (respectively, } ETV^+(C)\text{) and } C[B/X] \in \mathbf{NF} \text{ (respectively, } \mathbf{PF}\text{).} \end{aligned}$$

Hence, $B \in \mathbf{PF}$ by induction hypothesis. \square

Proposition 8.25. 1. If $X \in ETV^+(A)$ and $A[B/X] \in \mathbf{NF}$, then $B \in \mathbf{NF}$.

2. If $X \in ETV^-(A)$ and $A[B/X] \in \mathbf{PF}$, then $B \in \mathbf{NF}$.

Proof. The proof is quite parallel to the one for Proposition 8.24. Note that $B \in \mathbf{NF}$ if $B \sim \top$ by Definition 8.18. \square

Proposition 8.26. $A \in \mathbf{PF}$ (respectively, \mathbf{NF}) if and only if $A^c \in \mathbf{PF}$ (respectively, \mathbf{NF}).

Proof. By induction on $h(A)$, and by cases on the form of A . Note that the claim obviously holds when A is a \top -variant. Hence, we only consider the case that A is not.

Case: $A = X$ for some X . Trivial since $A^c = A$ in this case.

Case: $A = \bullet B$ for some B . In this case, $A \in \mathbf{PF}$ (respectively, \mathbf{NF}) iff $B \in \mathbf{PF}$ (respectively, \mathbf{NF}). On the other hand, $B \in \mathbf{PF}$ (respectively, \mathbf{NF}) iff $B^c \in \mathbf{PF}$ (respectively, \mathbf{NF}) by induction hypothesis. Therefore, it suffices to show that $A^c \in \mathbf{PF}$ (respectively, \mathbf{NF}) iff $B^c \in \mathbf{PF}$ (respectively, \mathbf{NF}), which can be established by considering the form of B^c as follows. Note that $B^c \neq \top$ since A is not a \top -variant. If $B^c = \bullet^n X$ for some n and X , then $A^c = \bullet B^c$, and if $B^c = \bullet^n (C \rightarrow D)$ for some n , C and D , where $n = 0$ is case of \simeq , then $A^{c\simeq} = \bullet B^{c\simeq}$ and $A^{c\simeq} = \bullet C \rightarrow \bullet D$. In either case, we can get $A^c \in \mathbf{PF}$ (respectively, \mathbf{NF}) iff $B^c \in \mathbf{PF}$ (respectively, \mathbf{NF}), by Definition 8.18.

Case: $A = B \rightarrow C$ for some B and C . Trivial since $A^c = A$ in this case.

Case: $A = \mu X.B$ for some X and B . Note that $A^c = B^c[A/X]$ in this case. Since A is not a \top -variant, B is not either; and hence, $B \not\sim \top$ by Theorem 4.14. First, we show the “if” part. Suppose that $A^c = B^c[A/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}), which also implies $B^c \in \mathbf{PF}$ (respectively, \mathbf{NF}) by Proposition 8.21. Hence, by induction hypothesis,

$$B \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}\text{).} \quad (39)$$

Furthermore, if $X \in ETV^-(B)$, then $X \in ETV^-(B^c)$ by Propositions 4.19 and 4.11.1; and therefore, $A \in \mathbf{NF}$ (respectively, \mathbf{PF}) by Proposition 8.25 (respectively, 8.24), which also implies $B \in \mathbf{NF}$ (respectively, $A \in \mathbf{PF}$) by Definition 8.18. That is,

$$B \in \mathbf{NF} \text{ (respectively, } A \in \mathbf{PF}\text{), if } X \in ETV^-(B). \quad (40)$$

On the other hand, $A \in \mathbf{TF}$ by Theorem 8.11 since A is not a \top -variant. Therefore, $A \in \mathbf{PF}$ (respectively, \mathbf{NF}) from (39) and (40) by Definition 8.18.

As for the “only if” part, suppose that $A \in \mathbf{PF}$ (respectively, \mathbf{NF}), which implies $B \in \mathbf{PF}$ (respectively, \mathbf{NF}). Hence, by induction hypothesis,

$$B^c \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}\text{).} \quad (41)$$

On the other hand, if $X \in ETV^-(B^c)$, then $X \in ETV^-(B)$ by Propositions 4.19 and 4.11.1; and therefore, $B \in \mathbf{NF}$ (respectively, $A \in \mathbf{PF}$) by Definition 8.18 since $A \in \mathbf{PF}$ (respectively, \mathbf{NF}), which also implies $A \in \mathbf{NF}$ (respectively, \mathbf{PF}). That is,

$$A \in \mathbf{NF} \text{ (respectively, } \mathbf{PF}\text{), if } X \in ETV^-(B^c). \quad (42)$$

Therefore, since $A \in \mathbf{PF}$ (respectively, \mathbf{NF}), we get $A^c = B^c[A/X] \in \mathbf{PF}$ (respectively, \mathbf{NF}) from (41) and (42) by Proposition 8.23. \square

Lemma 8.27. *If $A \in \mathbf{PF}$ (respectively, \mathbf{NF}), then A is positively (respectively, negatively) finite.*

Proof. By Theorem 8.11, it suffices to derive $C \in \mathbf{TF}$ from the following assumptions:

- (a) $A \in \mathbf{PF}$ (respectively, \mathbf{NF}),
- (b) $A \sim B[C/X]$,
- (c) $X \in ETV^+(B)$ (respectively, $ETV^-(B)$), and
- (d) $X \notin ETV^-(B)$ (respectively, $ETV^+(B)$).

We can assume that A and B are canonical without loss of generality, because $A^c \in \mathbf{PF}$ (respectively, \mathbf{NF}) iff $A \in \mathbf{PF}$ (respectively, \mathbf{NF}) by Proposition 8.26, and because $A^c \sim A$, $B^c[C/X] \sim B[C/X]$ and $ETV^\pm(B^c) = ETV^\pm(B)$ by Propositions 4.19, 4.4 and 4.11.1. Note that $B \neq \top$ from (c) by Proposition 2.9.5, and that Condition (c) can be rewritten as

$$(c') \quad dp_{\rightarrow}^+(B, X) < \infty \text{ (respectively, } dp_{\rightarrow}^-(B, X) < \infty)$$

by Proposition 2.15.1. Furthermore, if $A = \top$, then $A \notin \mathbf{PF}$ by Proposition 8.19 and Theorem 8.11 (respectively, $X \in ETV^+(B)$ from (b) by Propositions 2.10.2, 2.9.4 and Theorem 4.14 since $B \not\sim \top$). That is, $A = \top$ contradicts (a) (respectively, (d)). Hence, we can also assume that $A \neq \top$; and hence, $A \in \mathbf{TF}$ by Theorem 8.11. The proof proceeds by induction on $dp_{\rightarrow}^\pm(B, X)$, and by cases on the form of B .

Case: $B = \bullet^n Y$ for some n and Y . In this case, $X \notin ETV^-(B)$ and $Y = X$ from (c); and hence, $A \sim B[C/X] = \bullet^n C$. Since $A \in \mathbf{TF}$, A is tail finite by Theorem 8.11; and hence, so is C by Definition 8.5 and Proposition 4.29. Therefore, $C \in \mathbf{TF}$ by Theorem 8.11 again.

Case: $B = \bullet^n (D \rightarrow E)$ for some n , D and E such that $E \not\sim \top$, where $n = 0$ in case of \simeq . In this case, $A \sim B[C/X] = \bullet^n (D[C/X] \rightarrow E[C/X])$ and $B[C/X]^c = \bullet^n (D[C/X] \rightarrow E[C/X])$. Hence, $A = \bullet^n (F \rightarrow G)$ for some F and G such that $F \sim D[C/X]$ and $G \sim E[C/X] \not\sim \top$ by Proposition 4.21. Therefore, since $ETV^\pm(B) = ETV^\mp(D) \cup ETV^\pm(E)$ and $dp_{\rightarrow}^\pm(B, X) = \min(dp_{\rightarrow}^\mp(D, X), dp_{\rightarrow}^\pm(E, X)) + 1$, the assumptions (a), (c') and (d) imply the following.

$$\begin{aligned} dp_{\rightarrow}^-(D, X) < dp_{\rightarrow}^+(B, X) \text{ or } dp_{\rightarrow}^+(E, X) < dp_{\rightarrow}^+(B, X) \\ & \text{(respectively, } dp_{\rightarrow}^+(D, X) < dp_{\rightarrow}^-(B, X) \text{ or } dp_{\rightarrow}^-(E, X) < dp_{\rightarrow}^-(B, X)), \\ F \in \mathbf{NF} \text{ (respectively, } \mathbf{PF}\text{), } F \sim D[C/X] \text{ and } X \notin ETV^+(D) \text{ (respectively, } ETV^-(D)\text{), and} \\ G \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}\text{), } G \sim E[C/X] \text{ and } X \notin ETV^-(E) \text{ (respectively, } ETV^+(E)\text{).} \end{aligned}$$

Therefore, we get $C \in \mathbf{TF}$ by induction hypothesis. \square

Lemma 8.28. *If A is positively (respectively, negatively) finite, then $A \in \mathbf{PF}$ (respectively, \mathbf{NF}).*

Proof. We use Theorems 4.14 and 8.11 without specific mention, below. Suppose that A is positively (negatively) finite, i.e., $C \in \mathbf{TF}$ if there exists some B such that

$$A \sim B[C/X], \quad (43)$$

$$X \in ETV^+(B) \text{ (respectively, } ETV^-(B)\text{), and} \quad (44)$$

$$X \notin ETV^-(B) \text{ (respectively, } ETV^+(B)\text{).} \quad (45)$$

The proof proceeds by induction on $h(A)$, and by cases on the form of A .

Case: $A = Y$ for some Y . $A \in \mathbf{PF} \cap \mathbf{NF}$ by Definition 8.18.

Case: $A = \bullet D$ for some D . By induction hypothesis and Definition 8.18, it suffices to show that D is positively (respectively, negatively) finite. Suppose that $D \sim B'[C/X]$, $X \in ETV^+(B')$ and $X \notin ETV^-(B')$ (respectively, $X \in ETV^-(B')$ and $X \notin ETV^+(B')$) for some X and B' . Let $B = \bullet B'$. Then we get (43) through (45); and hence, $C \in \mathbf{TF}$.

Case: $A = D \rightarrow E$ for some D and E . By induction hypothesis and Definition 8.18, it suffices to show that (a) D is negatively (respectively, positively) finite, and (b) E is positively (respectively, negatively) finite. For (a), suppose that $D \sim B'[C/X]$, $X \in ETV^-(B')$ and $X \notin ETV^+(B')$ (respectively, $X \in ETV^+(B')$ and $X \notin ETV^-(B')$) for some X and B' . We can assume that $X \notin FTV(E)$ without loss of generality. Let $B = B' \rightarrow E$. Then we get (43) through (45); and hence, $C \in \mathbf{TF}$. We can similarly establish (b).

Case: $A = \mu Y.D$ for some Y and D . If $A \sim \top$, then obviously $A \in \mathbf{NF}$; and $A \sim X[\top/X]$, which means that A is not positively finite. Therefore, we now assume that $A \not\sim \top$, that is, A is not a \top -variant and $A \in \mathbf{TF}$. It suffices to show the following.

$$\mu Y.D \in \mathbf{TF} \text{ (respectively, } \mathbf{TExp}), \quad (46)$$

$$D \in \mathbf{PF} \text{ (respectively, } \mathbf{NF}), \text{ and} \quad (47)$$

$$Y \notin ETV^-(D) \text{ or } D \in \mathbf{NF} \text{ (respectively, } \mu Y.D \in \mathbf{PF}). \quad (48)$$

First, if A is positively finite, then $\mu Y.D \in \mathbf{TF}$ since $A \sim X[\mu Y.D/X]$. Thus, we get (46). Second, for (47), by induction hypothesis, it is sufficient to show that D is positively (respectively, negatively) finite. To this end, suppose that the following hold for some X and B' .

$$D \sim B'[C/X] \quad (49)$$

$$X \in ETV^+(B') \text{ (respectively, } ETV^-(B')), \text{ and} \quad (50)$$

$$X \notin ETV^-(B') \text{ (respectively, } ETV^+(B')). \quad (51)$$

We show that (49) through (51) imply $C \in \mathbf{TF}$. We can assume that $X \notin FTV(A) \cup \{Y\}$ without loss of generality. Let $B_1 = B'[A/Y]$. Then we get

$$\begin{aligned} A &\sim \mu Y.B'[C/X] && \text{(by (49) and Proposition 4.12.1)} \\ &\sim B'[C/X][\mu Y.B'[C/X]/Y] && \text{(by } (\cong\text{-fix})) \\ &\sim B'[C/X][A/Y] && \text{(by Proposition 4.4 since } A \sim \mu Y.B'[C/X]) \\ &= B'[C[A/Y]/X, A/Y] && \text{(since } X \neq Y) \\ &= B'[A/Y][C[A/Y]/X] && \text{(since } X \notin FTV(A)) \\ &= B_1[C[A/Y]/X]. \end{aligned}$$

Furthermore, since A is not a \top -variant, we get

$$X \in ETV^+(B_1) \text{ (respectively, } ETV^-(B_1))$$

from (50) by Proposition 2.11.1, and

$$X \notin ETV^-(B_1) \text{ (respectively, } ETV^+(B_1))$$

from (51) by Proposition 2.11.2 since $X \notin FTV(A)$. Therefore, $C[A/Y] \in \mathbf{TF}$ since A is positively (respectively, negatively) finite; and hence, $C \in \mathbf{TF}$ by Proposition 8.9.3. Thus, we establish that D is positively (respectively, negatively) finite; and hence, we get (47) by induction hypothesis.

What is left is to establish (48). Suppose that $Y \in ETV^-(D)$. Then, by Proposition 8.20, there exist some D' and Y' such that

$$D \sim D'[Y/Y'], \quad (52)$$

$$Y' \in ETV^-(D'), \text{ and} \quad (53)$$

$$Y' \notin ETV^+(D'). \quad (54)$$

We can assume that $Y' \notin FTV(D) \cup \{Y\}$ without loss of generality. Since D is proper in Y , so is $D'[Y/Y']$ from (52) by Proposition 4.11.2. As an important step for establishing (48), we show that D is negatively (respectively, positively) finite. To this end, suppose also that there exist some B'' and X such that

$$D \sim B''[C/X], \quad (55)$$

$$X \in ETV^-(B'') \text{ (respectively, } ETV^+(B'')), \text{ and} \quad (56)$$

$$X \notin ETV^+(B'') \text{ (respectively, } ETV^-(B'')). \quad (57)$$

It suffices to show that $C \in \mathbf{TF}$ follows from (55), (56) and (57). We can assume that $X \notin FTV(A) \cup FTV(D') \cup \{Y, Y'\}$. Therefore,

$$\begin{aligned} A &\sim \mu Y.D'[Y/Y'] && \text{(by (52) and Proposition 4.12.1)} \\ &\sim \mu Y.D'[D'[Y/Y']/Y'] && \text{(by Proposition 4.12.2)} \\ &\sim \mu Y.D'[D/Y'] && \text{(by (52) and Proposition 4.4)} \\ &\sim \mu Y.D'[B''[C/X]/Y'] && \text{(by (55) and Proposition 4.4)} \\ &\sim D'[B''[C/X]/Y'][\mu Y.D'[B''[C/X]/Y']/Y] && \text{(by } (\cong\text{-fix}) \\ &\sim D'[B''[C/X]/Y'][A/Y] && \text{(by Proposition 4.4 since } A \sim \mu Y.D'[B''[C/X]/Y']) \\ &= D'[B''[C/X][A/Y]/Y', A/Y] && \text{(since } Y' \neq Y) \\ &= D'[B''[A/Y][C[A/Y]/X]/Y', A/Y] && \text{(since } X \notin FTV(A) \cup \{Y\}) \\ &= D'[B''[A/Y]/Y', A/Y][C[A/Y]/X] && \text{(since } X \notin FTV(A) \cup FTV(D')) \\ &= D'[B''/Y'][A/Y][C[A/Y]/X] && \text{(since } Y' \neq Y) \end{aligned}$$

We get $X \in ETV^+(D'[B''/Y'])$ (respectively, $ETV^-(D'[B''/Y'])$) from (53) and (56) by Proposition 2.12.2; and hence,

$$X \in ETV^+(D'[B''/Y'][A/Y]) \text{ (respectively, } ETV^-(D'[B''/Y'][A/Y]))$$

by Proposition 2.11.1 since $Y \neq X$ and A is not a \mathbf{T} -variant. On the other hand, since $X \notin FTV(D')$, $X \notin ETV^-(D'[B''/Y'])$ (respectively, $ETV^+(D'[B''/Y'])$) from (54) and (57) by Proposition 2.12.3. Hence,

$$X \notin ETV^-(D'[B''/Y'][A/Y]) \text{ (respectively, } ETV^+(D'[B''/Y'][A/Y]))$$

by Proposition 2.11.2 because $X \notin FTV(A)$. Therefore, $C[A/Y] \in \mathbf{TF}$ since A is positively (respectively, negatively) finite; and hence, $C \in \mathbf{TF}$ by Proposition 8.9.3. Thus, D is negatively (respectively, positively) finite.

Now we can finally establish (48). In fact, we get $D \in \mathbf{NF}$ (respectively, \mathbf{PF}) by induction hypothesis, since D is negatively (positively) finite. Furthermore, in the case that A is negatively finite, $D \in \mathbf{PF}$ also implies $\mu Y.D \in \mathbf{PF}$, because $\mu Y.D \in \mathbf{TF}$ by assumption, and because $D \in \mathbf{NF}$ is already established as (47). \square

Proposition 8.29. *A is positively (negatively) finite if and only if $A \in \mathbf{PF}$ (\mathbf{NF}).*

Proof. Straightforward from Lemmas 8.27 and 8.28. \square

Thus, we verified that the notions of positively finiteness and negatively finiteness do not depend on the choice of \sim . Now we can proceed to the main result of this subsection, namely, the fact that every λ -term of positively finite types is maximal. Before proving that, we have to prepare the following three lemmas.

Lemma 8.30. *Let A and B be type expressions such that $A \preceq B$.*

1. *If B is positively finite, then so is A .*
2. *If A is negatively finite, then so is B .*

Proof. By Propositions 2.15.1, 8.29 and Theorem 8.11, it suffices to show that $D \in \mathbf{TF}$ if there exist some X, A, B and C such that either

- (a) $dp_{\rightarrow}^+(C, X) < \infty$, $dp_{\rightarrow}^-(C, X) = \infty$, $C[D/X] \preceq B$ and $B \in \mathbf{PF}$, or
- (b) $dp_{\rightarrow}^-(C, X) < \infty$, $dp_{\rightarrow}^+(C, X) = \infty$, $A \preceq C[D/X]$ and $A \in \mathbf{NF}$.

Suppose that (a) or (b) holds. We can assume that C is canonical with respect to \simeq , since $ETV^\pm(C) = ETV^\pm(C^{c\approx})$, $C[D/X] \simeq C^{c\approx}[D/X]$ and $dp_{\rightarrow}^\pm(C, X) = dp_{\rightarrow}^\pm(C^{c\approx}, X)$ by Propositions 4.19, 4.11.1, 4.4 and 4.10. The proof proceeds by induction on $dp_{\rightarrow}(C, X)$, and by cases on the form of C . We use Theorem 4.14, Propositions 2.3, 4.19 and 5.6 without mention, below.

Case: $C = \top$. This case is impossible, since $C = \top$ implies $ETV^\pm(C) = \{\}$ by Definition 2.7, which then implies $dp_{\rightarrow}^\pm(C, X) = \infty$ by Proposition 2.15.1.

Case: $C = \bullet^n Y$ for some n and Y . Note that (b) does not hold since $dp_{\rightarrow}^-(C, X) = \infty$ in this case. If (a) holds, then $Y = X$ since $dp_{\rightarrow}^+(C, X) < \infty$, and $B \not\preceq \top$ by Proposition 8.19 and Theorem 8.11. Hence, $C[D/X] = \bullet^n D \not\preceq \top$ since $C[D/X] \preceq B$ in this case, which implies $D \not\preceq \top$ by Proposition 4.3.1. Therefore, $D \in \mathbf{TF}$ by Theorem 8.11.

Case: $C = E \rightarrow F$ for some E and F such that $F \not\preceq \top$. In this case, $C[D/X] = (E \rightarrow F)[D/X] = E[D/X] \rightarrow F[D/X]$. Hence, if (a) holds, we get the following:

$$dp_{\rightarrow}^-(E, X) < dp_{\rightarrow}^+(C, X) \text{ or } dp_{\rightarrow}^+(F, X) < dp_{\rightarrow}^+(C, X), \quad (58)$$

$$dp_{\rightarrow}^+(E, X) = dp_{\rightarrow}^-(F, X) = \infty, \text{ and} \quad (59)$$

$$E[D/X] \rightarrow F[D/X] \preceq B, \quad (60)$$

since $dp_{\rightarrow}^\pm(C, X) = dp_{\rightarrow}^\pm(E \rightarrow F) = \min(dp_{\rightarrow}^\pm(E, X), dp_{\rightarrow}^\pm(F, X)) + 1$ by Definition 2.14. On the other hand, we get $B \not\preceq \top$ from $B \in \mathbf{PF}$ similarly to the previous case. Therefore, by (60) and Proposition 5.8, there exist some k, G and H such that

$$(a1) \ B^{c\approx} = G \rightarrow H,$$

$$(a2) \ G \preceq \bullet^k E[D/X] \text{ and } \bullet^k F[D/X] \preceq H.$$

Since $B \in \mathbf{PF}$, we get $B^{c\approx} \in \mathbf{PF}$ by Proposition 8.29 and Definition 8.17; and hence, $G \in \mathbf{NF}$ and $H \in \mathbf{PF}$ by Definition 8.18. Therefore, $D \in \mathbf{TF}$ from (a2), (58) and (59) by induction hypothesis, since $dp_{\rightarrow}^\pm(\bullet^k E, X) = dp_{\rightarrow}^\pm(E, X)$ and $dp_{\rightarrow}^\pm(\bullet^k F, X) = dp_{\rightarrow}^\pm(F, X)$.

On the other hand, if (b) holds, then we similarly get

$$dp_{\rightarrow}^+(E, X) < dp_{\rightarrow}^-(C, X) \text{ or } dp_{\rightarrow}^-(F, X) < dp_{\rightarrow}^-(C, X), \quad (61)$$

$$dp_{\rightarrow}^-(E, X) = dp_{\rightarrow}^+(F, X) = \infty \text{ and} \quad (62)$$

$$A \preceq E[D/X] \rightarrow F[D/X]. \quad (63)$$

We get $X \notin ETV^+(F)$ from $dp_{\rightarrow}^+(F, X) = \infty$ by Proposition 2.15.1; and hence, we get $F[D/X] \not\preceq \top$ and $E[D/X] \rightarrow F[D/X] \not\preceq \top$ from $F \not\preceq \top$ by Propositions 2.10.2 and 2.9.4. Therefore, by (63) and Proposition 5.8, there exist some k, G and H such that

$$(b1) \ A^{c\approx} = G \rightarrow H,$$

$$(b2) \ E[D/X] \preceq \bullet^k G \text{ and } \bullet^k H \preceq F[D/X].$$

We get $A^{c\approx} \in \mathbf{NF}$ from $A \in \mathbf{NF}$ by Proposition 8.29 and Definition 8.17. Furthermore, $H \not\preceq \top$ from (b2) since $F[D/X] \not\preceq \top$; and hence, $A \not\preceq \top$ by (b1). Therefore, $G \in \mathbf{PF}$ and $H \in \mathbf{NF}$ by Definition 8.18; and hence, $\bullet^k G \in \mathbf{PF}$ and $\bullet^k H \in \mathbf{NF}$. Therefore, $D \in \mathbf{TF}$ from (b2), (61) and (62) by induction hypothesis. \square

Lemma 8.31. *Let n be a non-negative integer, and x_1, x_2, \dots, x_n distinct individual variables such that $\{x_1, x_2, \dots, x_n\} \cap \text{Dom}(\Gamma) = \{\}$. If $\Gamma \vdash \lambda x_1. \lambda x_2. \dots \lambda x_n. M : A$ is derivable in $\lambda \mathbf{A}$, then there exist some m, B_1, B_2, \dots, B_n and C such that*

$$\begin{aligned} & \bullet^m \Gamma \cup \{x_1 : B_1, x_2 : B_2, \dots, x_n : B_n\} \vdash M : C \text{ is also derivable in } \lambda \mathbf{A}, \text{ and} \\ & B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_n \rightarrow C \preceq \bullet^m A. \end{aligned}$$

Proof. Proof proceeds by induction on n . It is trivial if $n = 0$. Hence, suppose that $\Gamma \vdash \lambda x_1. \lambda x_2. \dots \lambda x_n. M : A$ is derivable for some $n > 0$. By Lemma 7.7, for some k, B'_1 and D ,

$$\bullet^k \Gamma \cup \{x_1 : B'_1\} \vdash \lambda x_2. \dots \lambda x_n. M : D \text{ is derivable, and} \quad (64)$$

$$B'_1 \rightarrow D \preceq \bullet^k A. \quad (65)$$

Therefore, by applying the induction hypothesis to (64), for some l, B_2, B_3, \dots, B_n and C ,

$$\bullet^{k+l} \Gamma \cup \{x_1 : \bullet^l B'_1, x_2 : B_2, \dots, x_n : B_n\} \vdash M : C \text{ is derivable, and} \quad (66)$$

$$B_2 \rightarrow \dots \rightarrow B_n \rightarrow C \preceq \bullet^l D. \quad (67)$$

Let $m = k + l$ and $B_1 = \bullet^l B'_1$. Then, (66) means that $\bullet^m \Gamma \cup \{x_1 : B_1, x_2 : B_2, \dots, x_n : B_n\} \vdash M : C$ is derivable, and we get

$$\begin{aligned} B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_n \rightarrow C &= \bullet^l B'_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_n \rightarrow C \\ &\preceq \bullet^l B'_1 \rightarrow \bullet^l D && \text{(by (67))} \\ &\simeq \bullet^l (B'_1 \rightarrow D) && \text{(by } (\simeq\text{-K/L}) \text{)} \\ &\preceq \bullet^{k+l} A && \text{(by (65))} \\ &= \bullet^m A. \end{aligned} \quad \square$$

Note that the $(\simeq\text{-K/L})$ -rule is crucial to the proof of this lemma. However, if we remove the (shift)-rule from $\lambda\mathbf{A}$, n in the statement of Lemma 7.7, and hence, m of this lemma, can be 0, and by which the $(\simeq\text{-K/L})$ -rule becomes optional.

Lemma 8.32. *If $\bullet A \simeq B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_n \rightarrow C \not\preceq \top$, then $A \simeq B'_1 \rightarrow B'_2 \rightarrow B'_3 \rightarrow \dots \rightarrow B'_n \rightarrow C'$ for some $B'_1, B'_2, B'_3, \dots, B'_n$ and C' such that $\bullet B'_i \simeq B_i$ for every $i \in \{1, 2, 3, \dots, n\}$ and $\bullet C' \simeq C$.*

Proof. By induction on n . It is trivial in case of $n = 0$. Suppose that $\bullet A \simeq B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_n \rightarrow C$ for some $n > 0$. By Proposition 4.23.2, we get $A \simeq B'_1 \rightarrow A'$ for some B'_1 and A' such that $\bullet B'_1 \simeq B_1$ and $\bullet A' \simeq B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_n \rightarrow C$. Note that $A' \not\preceq \top$ since $A \not\preceq \top$. Therefore, by induction hypothesis, $A' \simeq B'_2 \rightarrow B'_3 \rightarrow \dots \rightarrow B'_n \rightarrow C'$ for some B'_2, B'_3, \dots, B'_n and C' such that $\bullet B'_i \simeq B_i$ for every $i \in \{2, 3, \dots, n\}$ and $\bullet C' \simeq C$. Thus, we get $A \simeq B'_1 \rightarrow B'_2 \rightarrow B'_3 \rightarrow \dots \rightarrow B'_n \rightarrow C'$. \square

Lemma 8.33. *Suppose that $A \not\preceq \top$. If $\Gamma \vdash x N_1 N_2 \dots N_n : A$ is derivable in $\lambda\mathbf{A}$, then for some $k, l, B_1, B_2, B_3, \dots, B_n$ and C ,*

- (a) $\Gamma(x) \simeq B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_n \rightarrow C$,
- (b) $\bullet^k C \preceq \bullet^l A$, and
- (c) $\bullet^l \Gamma \vdash N_i : \bullet^k B_i$ is derivable in $\lambda\mathbf{A}$ for every $i \in \{1, 2, \dots, n\}$.

Proof. By induction on the derivation of $\Gamma \vdash x N_1 N_2 \dots N_n : A$, and by cases on the last rule used in the derivation. Suppose that $\Gamma \vdash x N_1 N_2 \dots N_n : A$ is derivable in $\lambda\mathbf{A}$.

Case: (var). In this case, $n = 0$, and the derivation has the form of

$$\frac{}{\Gamma' \cup \{x : A\} \vdash x : A} \text{ (var)}$$

for some Γ' such that $\Gamma = \Gamma' \cup \{x : A\}$. Therefore, we get (a) through (c) by taking C, k, l as $C = A$ and $k = l = 0$.

Case: (shift). The derivation ends with

$$\frac{\bullet \Gamma \vdash x N_1 N_2 \dots N_n : \bullet A}{\Gamma \vdash x N_1 N_2 \dots N_n : A} \text{ (shift)}.$$

Note that $\bullet A \not\vdash \top$ since $A \not\vdash \top$. By induction hypothesis, for some $k', l', B'_1, B'_2, \dots, B'_n$ and C' ,

$$\begin{aligned} \bullet \Gamma(x) &\simeq B'_1 \rightarrow B'_2 \rightarrow B'_3 \rightarrow \dots \rightarrow B'_n \rightarrow C', \\ \bullet^{k'} C' &\preceq \bullet^{l'} \bullet A, \text{ and} \end{aligned} \quad (68)$$

$$\bullet^{l'} \bullet \Gamma \vdash N_i : \bullet^{k'} B'_i \text{ is derivable for every } i \ (1 \leq i \leq n). \quad (69)$$

Therefore, by Lemma 8.32, (a) holds for some B_1, B_2, \dots, B_n and C such that

$$\bullet B_i \simeq B'_i \text{ for every } i \ (1 \leq i \leq n), \text{ and} \quad (70)$$

$$\bullet C \simeq C'. \quad (71)$$

Taking k and l as $k = k' + 1$ and $l = l' + 1$, respectively, we get (b) from (68) and (71), and get (c) from (69) and (70).

Case: (\top) . This case is impossible since $A \not\vdash \top$.

Case: (\preceq) . In this case, the derivation ends with

$$\frac{\Gamma \vdash x N_1 N_2 \dots N_n : A' \quad A' \preceq A}{\Gamma \vdash x N_1 N_2 \dots N_n : A} (\preceq)$$

for some A' . Note that $A' \not\vdash \top$ by Proposition 5.6. By induction hypothesis, there exist some $k, l, B_1, B_2, B_3, \dots, B_n$ and C such that (a), $\bullet^k C \preceq \bullet^l A'$ and (c) hold, from the second of which we can also get (b) since $A' \preceq A$.

Case: $(\rightarrow I)$. This case is impossible by the form of $x N_1 N_2 \dots N_n$.

Case: $(\rightarrow E)$. In this case, $n > 0$, and the derivation ends with

$$\frac{\Gamma_1 \vdash x N_1 N_2 \dots N_{n-1} : D \rightarrow A \quad \Gamma_2 \vdash N_n : D}{\Gamma_1 \cup \Gamma_2 \vdash x N_1 N_2 \dots N_n : A} (\rightarrow E)$$

for some Γ_1, Γ_2 and D such that $\Gamma = \Gamma_1 \cup \Gamma_2$. Note that $D \rightarrow A \not\vdash \top$ by Proposition 4.15.3 since $A \not\vdash \top$. Hence, by induction hypothesis, for some $k', l, B_1, B_2, B_3, \dots, B_{n-1}$ and E ,

$$\Gamma_1(x) \simeq B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \dots \rightarrow B_{n-1} \rightarrow E, \quad (72)$$

$$\bullet^{k'} E \preceq \bullet^l (D \rightarrow A), \text{ and} \quad (73)$$

$$\bullet^l \Gamma_1 \vdash N_i : \bullet^{k'} B_i \text{ is derivable for every } i \ (1 \leq i < n). \quad (74)$$

Since $(\bullet^l (D \rightarrow A))^{c\approx} = \bullet^l D^{c\approx} \rightarrow \bullet^l A^{c\approx}$, by Propositions 5.8 and 4.19, there exist some B'_n, C' and l' such that

$$\bullet^{k'} E \simeq B'_n \rightarrow C', \quad (75)$$

$$\bullet^l D \preceq \bullet^{l'} B'_n \text{ and } \bullet^{l'} C' \preceq \bullet^l A \quad (76)$$

from (73). Note that $E \not\vdash \top$ by (73) since $D \rightarrow A \not\vdash \top$. Hence, by applying Proposition 4.23.2 to (75) repeatedly, there also exist some B_n and C such that

$$E \simeq B_n \rightarrow C, \quad (77)$$

$$\bullet^{k'} B_n \simeq B'_n \text{ and } \bullet^{k'} C \simeq C'. \quad (78)$$

We now get (a) from (72) and (77) since $\Gamma(x) = \Gamma_1(x)$. Let $k = k' + l'$. Then we get (b) from (76) and (78). On the other hand, $\bullet^l \Gamma_2 \vdash N_n : \bullet^l D$ is derivable by (nec) from $\Gamma_2 \vdash N_n : D$; and therefore, we get (c) from (74) by Proposition 7.2.2 and (\preceq) , since $\Gamma = \Gamma_1 \cup \Gamma_2$, $\bullet^{k'} B_i \preceq \bullet^k B_i$ for every $i \in \{1, 2, \dots, n-1\}$, and since $\bullet^l D \preceq \bullet^{l'} B'_n \simeq \bullet^{k'+l'} B_n = \bullet^k B_n$ from (76) and (78). \square

Theorem 8.34. *Let $\Gamma \vdash M : A$ be a derivable judgment of $\lambda\mathbf{A}$. If A is positively finite and $\Gamma(x)$ is negatively finite for every $x \in \text{Dom}(\Gamma)$, then M is maximal.*

Proof. We show that for every n , every node of the Böhm-tree of M at the level n is head normalizable, by induction on n . Since A is positively finite, M is head normalizable by Theorem 8.14, that is,

$$M \xrightarrow{\beta^*} \lambda x_1. \lambda x_2. \dots \lambda x_m. y N_1 N_2 \dots N_l$$

for some $x_1, x_2, \dots, x_m, y, N_1, N_2, \dots, N_l$. We can assume, without loss of generality, that x_1, x_2, \dots , and x_m are distinct individual variables such that $\{x_1, x_2, \dots, x_m\} \cap \text{Dom}(\Gamma) = \{\}$. It suffices to show that every node of the Böhm-tree of N_i at a level less than n is head normalizable for every i ($1 \leq i \leq l$). By Theorem 7.8, $\Gamma \vdash \lambda x_1. \lambda x_2. \dots \lambda x_m. y N_1 N_2 \dots N_l : A$ is also derivable in $\lambda\mathbf{A}$; and therefore, by Lemma 8.31, for some k, B_1, B_2, \dots, B_m and C ,

$$\bullet^k \Gamma \cup \{x_1 : B_1, x_2 : B_2, \dots, x_m : B_m\} \vdash y N_1 N_2 \dots N_l : C \text{ is derivable, and} \quad (79)$$

$$B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_m \rightarrow C \preceq \bullet^k A. \quad (80)$$

Note that since A is positively finite, so is $\bullet^k A$ by Proposition 8.29 and Definition 8.18; and hence, $B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_m \rightarrow C$ is also positively finite from (80) by Lemma 8.30. Therefore, C is positively finite, and B_1, B_2, \dots, B_m are negatively finite by Proposition 8.29 and Definition 8.18. Note also that $C \not\preceq \top$ since C is positively finite. Let $\Gamma' = \bullet^k \Gamma \cup \{x_1 : B_1, x_2 : B_2, \dots, x_m : B_m\}$. Note that $\Gamma'(x)$ is negatively finite for every $x \in \text{Dom}(\Gamma')$. By applying Lemma 8.33 to (79), for some $p, q, D_1, D_2, \dots, D_l$ and E ,

$$\Gamma'(y) \simeq D_1 \rightarrow D_2 \rightarrow \dots \rightarrow D_l \rightarrow E, \quad (81)$$

$$\bullet^p E \preceq \bullet^q C, \text{ and} \quad (82)$$

$$\bullet^q \Gamma' \vdash N_i : \bullet^p D_i \text{ is derivable for every } i \text{ } (1 \leq i \leq l). \quad (83)$$

Since $\bullet^q C$ is also positively finite, so is E from (82) by Lemma 8.30, Proposition 8.29 and Definition 8.18, i.e., $E \not\preceq \top$; and therefore, D_i is positively finite, and so is $\bullet^p D_i$, for every i ($1 \leq i \leq l$) from (81) because $\Gamma'(y)$ is negatively finite. Since $\bullet^q \Gamma'(x)$ is also negatively finite for every $x \in \text{Dom}(\Gamma')$, by applying induction hypothesis to (83), every node of the Böhm-tree of N_i at a level less than n is head normalizable for every i ($1 \leq i \leq l$); that is, so is one of M at a level less than or equal to n . \square

8.4. Simple types

By the standard method due to Tait[27], it can be also shown that λ -terms of simple types, namely, those having no occurrence of the modal operator, are normalizable.

Definition 8.35. We denote by \mathcal{N} the subset of normalizable λ -terms of \mathbf{Exp} , and define a subset \mathcal{K}_n of $\mathbf{Exp}/_{\beta}$ as follows.

$$\mathcal{K}_n = \{ [x N_1 N_2 \dots N_n] \mid x \in \mathbf{Var}, n \geq 0 \text{ and } N_i \in \mathcal{N} \text{ for every } i \text{ } (i = 1, 2, \dots, n) \}.$$

Lemma 8.36. *Consider the term model of \mathbf{Exp} , an arbitrary $\lambda\mathbf{A}$ -frame $\langle \mathcal{W}, \triangleright \rangle$ and a hereditary type environment η such that $\mathcal{K}_n \subseteq \eta(X)_p \subseteq \mathcal{N}$ for every type variable X and $p \in \mathcal{W}$. Let A be a type expression with no occurrence of the modal operator \bullet . Then, $\mathcal{K}_n \subseteq \mathcal{I}(A)_p^\eta \subseteq \mathcal{N}$ for every $p \in \mathcal{W}$.*

Proof. By induction on $h(A)$, and by cases on the form of A .

Case: $A = Y$ for some Y . Obvious since $\mathcal{I}(A)_p^\eta = \eta(Y)_p$ by Proposition 3.9.

Case: $A = B \rightarrow C$ for some B and C . First, we show $\mathcal{K}_n \subseteq \mathcal{I}(B \rightarrow C)_p^\eta$. Let $u = [x N_1 N_2 \dots N_n] \in \mathcal{K}_n$, and suppose that $p \not\triangleright q$ and $[L] \in \mathcal{I}(B)_q^\eta$. By induction hypothesis, $[L] \in \mathcal{N}$. Then, $u \cdot [L] = [x N_1 N_2 \dots N_n L] \in \mathcal{K}_n \subseteq \mathcal{I}(C)_q^\eta$ by induction hypothesis again; and hence, $u \in \mathcal{I}(B \rightarrow C)_p^\eta$ by Proposition 3.9. Second, to show $\mathcal{I}(B \rightarrow C)_p^\eta \subseteq \mathcal{N}$, suppose that $[M] \in \mathcal{I}(B \rightarrow C)_p^\eta$. Let y be a fresh individual variable. Since

$[y] \in \mathcal{K}_n \subseteq \mathcal{I}(B)_p^\eta$ by induction hypothesis, we get $[M] \cdot [y] = [My] \in \mathcal{I}(C)_p^\eta$ by Proposition 3.9; and hence, $[My] \in \mathcal{N}$ by induction hypothesis. Therefore, My has a normal form, say L . There are two possible cases: for some K , (i) $M \xrightarrow[\beta]{*} K$ and $L = Ky$, or (ii) $M \xrightarrow[\beta]{*} \lambda y. K$ and $K \xrightarrow[\beta]{*} L$. In either case, M obviously has a normal form.

Case: $A = \mu X. B$ for some X and B . In this case, $X \notin FTV(B)$ since B is proper in X , and since A has no occurrence of \bullet . Therefore, $\mathcal{I}(\mu X. B)_p^\eta = \mathcal{I}(B)_p^\eta$ by Proposition 3.9; and hence, obvious from induction hypothesis. \square

Theorem 8.37. *Suppose that the modal operator does not occur in Γ or A . If $\Gamma \vdash M : A$ is derivable in $\lambda\mathbf{A}$, then M is normalizable.*

Proof. Let $\Gamma = \{x_1 : B_1, x_2 : B_2, \dots, x_n : B_n\}$ and let $\langle \mathcal{W}, \triangleright \rangle$ be an arbitrary $\lambda\mathbf{A}$ -frame. Consider the term model $\langle \mathcal{V}, \cdot, \llbracket \cdot \rrbracket^\mathcal{V} \rangle$ of \mathbf{Exp} and a type environment η such that $\eta(X)_p = \mathcal{K}_n$ for every $X \in \mathbf{TVar}$ and $p \in \mathcal{W}$. Furthermore, let ρ be an individual environment such that $\rho(x) = [x]$ for every $x \in \mathbf{Var}$. For every $i \in \{1, 2, \dots, n\}$, since $\rho(x_i) = [x_i] \in \mathcal{K}_n$, we get $\rho(x_i) \in \mathcal{I}(B_i)_p^\eta$ for every $p \in \mathcal{W}$ by Lemma 8.36. Hence, $\llbracket M \rrbracket_\rho^\mathcal{V} = [M] \in \mathcal{I}(A)_p^\eta \subseteq \mathcal{N}$ by Theorem 7.3 and Lemma 8.36 again. \square

Finiteness of type expressions, namely, not including recursive types, is not sufficient for assuring the normalizability of inhabitants. In the proof of Lemma 8.36, we cannot show that $\mathcal{I}(\bullet B)_p^\eta \subseteq \mathcal{N}$ from the induction hypothesis $\mathcal{I}(B)_p^\eta \subseteq \mathcal{N}$, since it might be the case that $\mathcal{I}(\bullet B)_p^\eta \supsetneq \mathcal{I}(B)_p^\eta$. In fact, we have already observed that Curry's \mathbf{Y} has such a type $(\bullet X \rightarrow X) \rightarrow X$ in Example 6.4.

9. $\lambda\mathbf{A}$ as a basis for logic of programming

The typing system $\lambda\mathbf{A}$ can be easily extended to cover full propositional and second-order types. For example, we can add the following equality and typing rules for product types.

$$\bullet(A_1 \times A_2) \cong \bullet A_1 \times \bullet A_2$$

$$\frac{\Gamma_1 \vdash M : A_1 \quad \Gamma_2 \vdash N : A_2}{\Gamma_1 \cup \Gamma_2 \vdash \langle M, N \rangle : A_1 \times A_2} (\times I) \qquad \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \mathbf{p}_i M : A_i} (\times E) \quad (i = 1, 2)$$

However, even with such extensions, type expressions discussed so far can only refer to approximative worlds at a fixed distance from the current world, and do not have enough power to express the manner in which approximation proceeds in general recursive programs. In order to handle such programs, we have to extend our logic to a predicate logic, and allow type expressions such as $\bullet^t A$ as well-formed type expressions, with t being a numeric expression. We also need enough arithmetic endowing the typing system to discuss such t . With the help of such extensions, $\lambda\mathbf{A}$ can be a basis for logic of a wide range of programs. In this section we give some examples.

9.1. Recursive programs over non-negative integers

Provided such an extension to predicate logic endowed with an arithmetic, we can construct a wide range of recursive programs with fixed-point combinators ensuring their termination. For example, let $\mathbf{nat}(n)$ represent the type of (implementations of) the non-negative integer n , and suppose that f is a primitive recursive functional defined as

$$f x \equiv \text{if } (x = 0) \text{ then } c \text{ else } g(f(x - 1)) x,$$

where c is a program of a type $A(0)$, g a type $A(n - 1) \rightarrow \mathbf{nat}(n) \rightarrow A(n)$ for every positive integer n with a primitive recursive function \div defined in the arithmetic as

$$x \div y \equiv \begin{cases} x - y & (\text{if } x \geq y) \\ 0 & (\text{otherwise}) \end{cases}$$

We also assume that c or g includes no free occurrence of x , and that the infix operator $-$ is a program that satisfies $\forall m. \forall n. \mathbf{nat}(m) \rightarrow \mathbf{nat}(n) \rightarrow \mathbf{nat}(n \div m)$. We will show that f has the type $\forall n. \mathbf{nat}(n) \rightarrow \bullet^n A(n)$, where, in general, $\forall n. B(n)$ is interpreted as

$$\mathcal{I}(\forall n. B(n))_p^\eta = \{ u \mid u \in \mathcal{I}(B(n))_p^\eta \text{ for every non-negative integer } n \}.$$

By virtue of the fixed-point combinator \mathbf{Y} , we can reformulate the definition of f to

$$f \equiv \mathbf{Y} (\lambda f. \lambda x. \mathbf{if} (x = 0) \mathbf{then} c \mathbf{else} g (f (x - 1)) x).$$

Let $C = \forall n. \mathbf{nat}(n) \rightarrow \bullet^n A(n)$. It suffices to show that $\mathbf{if} (x = 0) \mathbf{else} g (f (x - 1)) x : \bullet^n A(n)$ assuming $f : \bullet C$ and $x : \mathbf{nat}(n)$, since $\mathbf{Y} : (\bullet C \rightarrow C) \rightarrow C$. If $n = 0$, then this is straightforward from $c : A(0)$. The case that $n > 0$ can be accomplished as follows.

$$\begin{array}{ll} f : \forall n. \bullet(\mathbf{nat}(n) \rightarrow \bullet^n A(n)) & (\text{since } \bullet \text{ is interchangeable with } \forall n) \\ f : \bullet(\mathbf{nat}(n \div 1) \rightarrow \bullet^{n-1} A(n \div 1)) & (\text{by instantiation, i.e., } (\forall E)) \\ f : \bullet \mathbf{nat}(n \div 1) \rightarrow \bullet^{n-1+1} A(n \div 1) & (\text{by } (\simeq\text{-}\mathbf{K}/\mathbf{L}) \text{ or } (\preceq\text{-}\mathbf{K})) \\ f : \bullet \mathbf{nat}(n \div 1) \rightarrow \bullet^n A(n \div 1) & (\text{since } n \div 1 + 1 = n) \\ f (x - 1) : \bullet^n A(n \div 1) & (\text{since } x - 1 : \bullet \mathbf{nat}(n \div 1)) \\ g (f (x - 1)) : \bullet^n \mathbf{nat}(n) \rightarrow \bullet^n A(n) & (\text{since } g : \bullet^n A(n \div 1) \rightarrow \bullet^n \mathbf{nat}(n) \rightarrow \bullet^n A(n)) \\ g (f (x - 1)) x : \bullet^n A(n) & (\text{since } x : \bullet^n \mathbf{nat}(n)) \end{array}$$

Thus, we get $\mathbf{if} (x = 0) \mathbf{then} c \mathbf{else} g (f (x - 1)) x : \bullet^n A(n)$.

In general, if a function f of a type $A(\vec{n}) \rightarrow B$ can be recursively defined based on a well-founded measure $\pi(\vec{n})$ on the input, then we can give f a type $A(\vec{n}) \rightarrow \bullet^{\pi(\vec{n})} B$. Consider the following recursive program, which represents McCarthy's 91-function.

$$f x \equiv \mathbf{if} (x > 100) \mathbf{then} x - 10 \mathbf{else} f (f (x + 11))$$

We can show that f has a type, or satisfies a specification, $\forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n))$, where n ranges over non-negative integers, and g is a primitive recursive function defined in the arithmetic as follows.

$$g(x) \equiv \begin{cases} x - 10 & (\text{if } x > 100) \\ 91 & (\text{otherwise}) \end{cases}$$

Suppose that $f : \bullet \forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n))$ and $x : \mathbf{nat}(n)$. The type of \mathbf{Y} assures that it suffices to show

$$\mathbf{if} (x > 100) \mathbf{then} x - 10 \mathbf{else} f (f (x + 11)) : \forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n)). \quad (84)$$

We assume that the infix operator $+$ is a program of the type $\forall m. \forall n. \mathbf{nat}(m) \rightarrow \mathbf{nat}(n) \rightarrow \mathbf{nat}(n + m)$. First, we get

$$f (x + 11) : \bullet \bullet^{101 \div (n+11)} \mathbf{nat}(g(n + 11)).$$

If $n \leq 90$, then we get $f (x + 11) : \bullet^{91 \div n} \mathbf{nat}(91)$ by the definitions of \div and g ; and therefore,

$$f (f (x + 11)) : \bullet^{91 \div n} \bullet^{101 \div 91} \mathbf{nat}(g(91)),$$

which type is equivalent to $\bullet^{101 \div n} \mathbf{nat}(91)$. On the other hand, if $90 < n \leq 100$, then we similarly get $f (x + 11) : \bullet \mathbf{nat}(n + 1)$; and therefore,

$$f (f (x + 11)) : \bullet \bullet^{101 \div (n+1)} \mathbf{nat}(g(n + 1)),$$

which type is also equivalent to $\bullet^{101 \div n} \mathbf{nat}(91)$. Otherwise, i.e., if $100 < n$, obviously $x - 10 : \mathbf{nat}(g(n))$. Thus, we establish (84).

In this derivation, the fixed-point combinator worked as the induction scheme discussed in Section 1 with a sequence $S_0, S_1, S_2, \dots, S_n$ as follows.

$$\begin{aligned} S_0 &= \mathcal{V} \\ S_{n+1} &= \{ f \mid \forall x \geq 101 \div n. f(x) = g(x) \} \quad (n = 0, 1, 2, \dots) \end{aligned}$$

The typing $\vdash f : \forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n))$ does not mean that the function f computes the output in $101 \div n$ steps. Actually, $f(n)$ involves $2(101 \div n)$ recursive calls. Recall that our interpretations of types are closed under \rightarrow_β . It only means that the output satisfies a specification that is at most $101 \div n$ steps *weaker*, in a sense which only the programmer knows, than the specification of the input. However, on the contrary, if a recursively defined program f over non-negative integers computes its output $g(n)$ within $\pi(n)$ levels of recursion for every input n , the typing judgment $\vdash f : \forall n. \mathbf{nat}(n) \rightarrow \bullet^{\pi(n)} \mathbf{nat}(g(n))$ should be derivable.

Note also that if we only consider $\lambda\mathbf{A}$ -frames such that

$$\text{for every } p \in \mathcal{W}, \text{ there exists some } q \in \mathcal{W} \text{ such that } q \triangleright p,^9 \quad (85)$$

and if $\mathcal{I}(\mathbf{nat}(n))_p^\eta$ does not depend on p , then the interpretation of $\vdash f : \forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n))$ implies $f \in \mathcal{I}(\forall n. \mathbf{nat}(n) \rightarrow \mathbf{nat}(g(n)))_p^\eta$ for every $p \in \mathcal{W}$. For, in such a frame, for each n , every possible world p has another world q from which p is accessible in more than $101 \div n$ steps. Because the typing judgment is valid in every possible world, we get $f \in \mathcal{I}(\mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n)))_q^\eta$ for such q , and which implies that $f \in \mathcal{I}(\mathbf{nat}(n) \rightarrow \mathbf{nat}(g(n)))_p^\eta$ provided that $\mathcal{I}(\mathbf{nat}(n))_p^\eta$ does not depend on p . Thus, we get $f \in \mathcal{I}(\mathbf{nat}(n) \rightarrow \mathbf{nat}(g(n)))_p^\eta$ for every $p \in \mathcal{W}$ and $n \in \mathbb{N}$. It can be also observed that $\vdash f : \forall n. \mathbf{nat}(n) \rightarrow \mathbf{nat}(g(n))$ becomes formally derivable from $\vdash f : \forall n. \mathbf{nat}(n) \rightarrow \bullet^{101 \div n} \mathbf{nat}(g(n))$ by introducing another modality, say $!$, which is interpreted as

$$\mathcal{I}(!A)_p^\eta = \{ u \mid u \in \mathcal{I}(A)_q^\eta \text{ for every } q \in \mathcal{W} \},$$

and accordingly enjoys the following subtyping relations and typing rules:

1. $A \preceq B$ implies $!A \preceq !B$
2. $!(A \rightarrow B) \preceq !A \rightarrow !B$
3. $!A \preceq A$
4. $!!A \cong !A$
5. $!\bullet^t A \cong !A$
6. $!\mathbf{nat}(n) \cong \mathbf{nat}(n)$

$$\frac{\Gamma_1 \vdash M : A}{!\Gamma_1 \cup \Gamma_2 \vdash M : !A} (!) \qquad \frac{!\Gamma_1 \cup \bullet\Gamma_2 \vdash M : \bullet A}{!\Gamma_1 \cup \Gamma_2 \vdash M : A} (\text{shift})$$

where the new (shift)-rule supersedes the original one in Definition 6.3. While the equality $!\bullet^t A \cong !A$ and the new (shift)-rule depend on the restriction (85), the other rules are valid for any frame. We might also need the (subst)-rule since Proposition 7.4.2 would not hold for the extended system. Recursive type variables are not allowed to occur in scopes of the $!$ -operator, and $\mathcal{I}(A)_p^\eta$ is now defined by induction on the lexicographic ordering of $\langle b(A), p, r(A) \rangle$, where $b(A)$ is the depth of nesting occurrences of $!$ in A .

⁹For example, the set of non-negative integers, or ordinals, and the “greater than” relation $>$ constitutes such a frame.

9.2. Infinite data structures

Streams, or infinite sequences, of data of a type X are representable by the type $\mu Y. X \times \bullet Y$. We can construct recursive programs for streams with fixed-point combinators such as Curry's \mathbf{Y} . For example, a program that generates a stream of a given constant of type X as follows, where $A = \mu Y. X \times \bullet Y$.

$$\begin{array}{c}
\frac{}{x : X \vdash x : X} \text{ (var)} \quad \frac{}{y : \bullet A \vdash y : \bullet A} \text{ (var)} \\
\hline
\frac{}{x : X, y : \bullet A \vdash \langle x, y \rangle : X \times \bullet A} \text{ (}\times\text{I)} \\
\vdots \\
\frac{}{x : X, y : \bullet A \vdash \langle x, y \rangle : A} \text{ (}\cong\text{)} \\
\hline
\frac{}{x : X \vdash \lambda y. \langle x, y \rangle : \bullet A \rightarrow A} \text{ (}\rightarrow\text{I)} \\
\hline
\frac{}{\vdash \mathbf{Y} : (\bullet A \rightarrow A) \rightarrow A} \text{ (}\rightarrow\text{E)} \\
\hline
\frac{}{x : X \vdash \mathbf{Y}(\lambda y. \langle x, y \rangle) : A} \text{ (}\rightarrow\text{I)} \\
\hline
\vdash \lambda x. \mathbf{Y}(\lambda y. \langle x, y \rangle) : X \rightarrow A
\end{array}$$

The following shows the derivation of a program that merges two streams, where $B = A \rightarrow A \rightarrow A$ and $\Gamma = \{f : \bullet B, x : A, y : A\}$.

$$\begin{array}{c}
\frac{}{\Gamma \vdash x : A} \text{ (var)} \quad \frac{}{\Gamma \vdash f : \bullet B} \text{ (var)} \quad \frac{}{\Gamma \vdash y : A} \text{ (var)} \\
\hline
\frac{}{\Gamma \vdash x : X \times \bullet A} \text{ (}\cong\text{)} \quad \frac{}{\Gamma \vdash f : \bullet A \rightarrow \bullet A \rightarrow \bullet A} \text{ (}\preceq\text{)} \quad \frac{}{\Gamma \vdash y : \bullet A} \text{ (}\preceq\text{)} \quad \frac{}{\Gamma \vdash x : X \times \bullet A} \text{ (}\cong\text{)} \\
\hline
\frac{}{\Gamma \vdash \mathbf{p}_1 x : X} \text{ (}\times\text{E)} \quad \frac{}{\Gamma \vdash f y : \bullet A \rightarrow \bullet A} \text{ (}\rightarrow\text{E)} \quad \frac{}{\Gamma \vdash \mathbf{p}_2 x : \bullet A} \text{ (}\times\text{E)} \\
\hline
\frac{}{\Gamma \vdash f y(\mathbf{p}_2 x) : \bullet A} \text{ (}\rightarrow\text{E)} \\
\hline
\frac{}{\Gamma \vdash \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : X \times \bullet A} \text{ (}\times\text{I)} \\
\hline
\frac{}{\Gamma \vdash \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : A} \text{ (}\cong\text{)} \\
\hline
\frac{}{f : \bullet B, x : A \vdash \lambda y. \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : A \rightarrow A} \text{ (}\rightarrow\text{I)} \\
\hline
\frac{}{f : \bullet B \vdash \lambda x. \lambda y. \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : B} \text{ (}\rightarrow\text{I)} \\
\hline
\frac{}{\vdash \lambda f. \lambda x. \lambda y. \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : \bullet B \rightarrow B} \text{ (}\rightarrow\text{I)} \\
\hline
\vdash \mathbf{Y} : (\bullet B \rightarrow B) \rightarrow B \quad \vdash \lambda f. \lambda x. \lambda y. \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle : \bullet B \rightarrow B \\
\hline
\vdash \mathbf{Y}(\lambda f. \lambda x. \lambda y. \langle \mathbf{p}_1 x, f y(\mathbf{p}_2 x) \rangle) : B
\end{array}$$

To capture more complicated recursion over streams, we naturally need an extension to predicate logic such as used in the case of the 91-function. For example, the prime number generator based on the sieve of Eratosthenes is derivable in our framework with such an extension. First, by using \mathbf{Y} , define three recursive programs **sieve**, **primes** and **enum** as follows.

$$\begin{aligned}
\mathbf{enum} \, x &\equiv \langle x, \mathbf{enum} \, (x + 1) \rangle \\
\mathbf{sieve} \, x \, w &\equiv \mathbf{if} \, ((\mathbf{p}_1 \, w) \bmod x = 0) \, \mathbf{then} \, \mathbf{sieve} \, x \, (\mathbf{p}_2 \, w) \, \mathbf{else} \, \langle \mathbf{p}_1 \, w, \mathbf{sieve} \, x \, (\mathbf{p}_2 \, w) \rangle \\
\mathbf{primes} \, w &\equiv \langle \mathbf{p}_1 \, w, \mathbf{primes} \, (\mathbf{sieve} \, (\mathbf{p}_1 \, w) \, (\mathbf{p}_2 \, w)) \rangle
\end{aligned}$$

Then, $\mathbf{primes} \, (\mathbf{enum} \, 2)$ generates the stream of all the prime numbers in ascending order, which can be formally verified by the extended typing system. Let m, n, k and l range over non-negative integers, and let $\mathit{preprime}$, prime , npp , and np be defined in the arithmetic as follows.

$$\begin{aligned}
\mathit{preprime}(m, n) &\equiv 1 < m \wedge \forall k. \forall l. 1 < k < m \wedge k < n \supset m \neq k \cdot l \\
\mathit{prime}(n) &\equiv \mathit{preprime}(n, n) \\
\mathit{npp}(m, n) &\equiv \min\{k \mid m \leq k \wedge \mathit{preprime}(k, n)\} \\
\mathit{np}(n) &\equiv \min\{k \mid n \leq k \wedge \mathit{prime}(k)\}
\end{aligned}$$

Note that $prime(n)$ means that n is a prime number, and $np(n)$ represents the least prime number that is greater than or equal to n . We use these two predicates and two functions for annotating type expression, and define three types **PPS**, **PS** and **Sieve** as follows.

$$\begin{aligned} \mathbf{PPS}(m, n) &\equiv \mathbf{nat}(m) \wedge preprime(m, n) \times \bullet \bullet^{npp(m+1, n) \div m \div 1} \mathbf{PPS}(npp(m+1, n), n) \\ \mathbf{PS}(n) &\equiv \mathbf{nat}(n) \wedge prime(n) \times \bullet \bullet^{np(n+1) \div n \div 1} \mathbf{PS}(np(n+1)) \\ \mathbf{Sieve}(m, n) &\equiv \mathbf{nat}(n) \wedge prime(n) \rightarrow \mathbf{PPS}(m, n) \rightarrow \\ &\quad \bullet^{npp(m, np(n+1)) \div m} \mathbf{PPS}(npp(m, np(n+1)), np(n+1)) \end{aligned}$$

The two recursive types, namely **PPS** and **PS**, are defined without explicit occurrence of μ for readability. The streams of all the prime numbers in ascending order should have the type **PS**(2). Actually, we can derive the following typing.

$$\begin{aligned} \mathbf{enum} &: \forall n. \mathbf{nat}(n) \wedge 1 < n \rightarrow \mathbf{PPS}(n, 2) \\ \mathbf{sieve} &: \forall m. \forall n. \mathbf{Sieve}(m, n) \\ \mathbf{primes} &: \forall n. \mathbf{PPS}(n, n) \rightarrow \mathbf{PS}(n) \\ \mathbf{primes}(\mathbf{enum} \ 2) &: \mathbf{PS}(2) \end{aligned}$$

As can be seen in this example, the intrinsic complexity of ensuring the convergence, or the productivity, of programs is not mitigated by use of the approximation modality. Its importance consists in the facts that (a) it enables us to discuss the convergence without going into the computational behavior of programs, and by which (b) it is possible to keep the modularity of programs. The derivations above can be done even if the left hand sides of “:” in the typing judgments are hidden, and through the proofs-as-programs notion, the derivation itself can be regarded as an executable program.

9.3. The **Nat**(n)-example

We now reconsider the example of object-oriented natural numbers with an addition method. We revise the definition of **Nat**(n) as follows.

$$\mathbf{Nat}(n) \equiv ((n = 0) + (n > 0 \wedge \bullet \mathbf{Nat}(n-1))) \times (\forall m. \bullet \mathbf{Nat}(m) \rightarrow \bullet \mathbf{Nat}(n+m))$$

Then, the specifications of **add** and **add'** are now different as follows.

$$\mathbf{add} : \forall n. \forall m. \mathbf{Nat}(n) \rightarrow \bullet \mathbf{Nat}(m) \rightarrow \bullet \mathbf{Nat}(n+m) \quad (86)$$

$$\mathbf{add}' : \forall n. \forall m. \bullet \mathbf{Nat}(n) \rightarrow \mathbf{Nat}(m) \rightarrow \bullet \mathbf{Nat}(n+m) \quad (87)$$

We can show $\mathbf{s} : \forall n. \mathbf{Nat}(n) \rightarrow \mathbf{Nat}(n+1)$ by deriving $\mathbf{s} \ x : \mathbf{Nat}(n+1)$ from the following assumptions.

$$\mathbf{s} : \bullet \forall n. \mathbf{Nat}(n) \rightarrow \mathbf{Nat}(n+1) \quad (88)$$

$$x : \mathbf{Nat}(n) \quad (89)$$

In fact, from $x : \mathbf{Nat}(n)$, we get

$$\mathbf{i}_2 \ x : (n+1 = 0) + (n+1 > 0 \wedge \bullet \mathbf{Nat}(n+1-1)). \quad (90)$$

Furthermore, if $y : \bullet \mathbf{Nat}(m)$, then

$$\begin{aligned} \mathbf{s} &: \bullet \mathbf{Nat}(m) \rightarrow \bullet \mathbf{Nat}(m+1) && \text{(by } (\forall E) \text{ from (88), and } (\preceq)) \\ \mathbf{s} \ y &: \bullet \mathbf{Nat}(m+1) && \text{(by } (\rightarrow E)) \\ \mathbf{add} \ x \ (\mathbf{s} \ y) &: \bullet \mathbf{Nat}(n+m+1) && \text{(by (86) and (89))} \\ &\cong \bullet \mathbf{Nat}(n+1+m) && \text{(since } n+m+1 = n+1+m). \end{aligned}$$

Hence,

$$\begin{aligned}
\lambda y. \mathbf{add} \ x \ (\mathbf{s} \ y) & : \bullet \mathbf{Nat} m \rightarrow \bullet \mathbf{Nat}(n+1+m) & (\text{by } (\rightarrow \mathbf{I})) \\
\lambda y. \mathbf{add} \ x \ (\mathbf{s} \ y) & : \forall m. \bullet \mathbf{Nat}(m) \rightarrow \bullet \mathbf{Nat}(n+1+m) & (\text{by } (\forall \mathbf{I})) \\
\langle \mathbf{i}_2 \ x, \lambda y. \mathbf{add} \ x \ (\mathbf{s} \ y) \rangle & : \mathbf{Nat}(n+1) & (\text{by } (\times \mathbf{I}) \text{ with } (90)).
\end{aligned}$$

Thus, we get $\mathbf{s} : \forall n. \mathbf{Nat}(n) : \mathbf{Nat}(n+1)$. However, on the other hand, under similar assumptions, we can only get

$$\mathbf{add}' \ x \ (\mathbf{s}' \ y) : \bullet \bullet \mathbf{Nat}(n+m+1),$$

from (87), and fail to derive $\mathbf{s}' : \forall n. \mathbf{Nat}(n) \rightarrow \mathbf{Nat}(n+1)$.

10. Modal logics behind $\lambda \mathbf{A}$

In this section, we consider $\lambda \mathbf{A}$ as a modal logic by ignoring left hand sides of “:” from typing judgments, and show that it corresponds to an intuitionistic version of the logic of provability \mathbf{GL} (cf. [4]). The modal logic \mathbf{GL} is also denoted by \mathbf{G} (for Gödel), \mathbf{L} (for Löb), \mathbf{PrL} , \mathbf{KW} , or $\mathbf{K4W}$, in the literature.

Definition 10.1 (Formal systems $\mathbf{miK4}$ and $\mathbf{LA}\mu$). Regarding type expressions as logical formulae, we define a formal system $\mathbf{miK4}$, the minimal and implication fragment of the modal logic $\mathbf{K4}$, by the following inference rules, where Γ denotes a finite set of formulae.

$$\begin{array}{c}
\frac{}{\Gamma \cup \{A\} \vdash A} \text{ (assump)} \qquad \frac{\Gamma_1 \vdash A}{\bullet \Gamma_1 \cup \Gamma_2 \vdash \bullet A} \text{ (nec)} \\
\\
\frac{\Gamma \vdash \bullet A}{\Gamma \vdash \bullet \bullet A} \text{ (4)} \qquad \frac{\Gamma \cup \{A\} \vdash B}{\Gamma \vdash A \rightarrow B} (\rightarrow \mathbf{I}) \qquad \frac{\Gamma_1 \vdash A \rightarrow B \quad \Gamma_2 \vdash A}{\Gamma_1 \cup \Gamma_2 \vdash B} (\rightarrow \mathbf{E})
\end{array}$$

Note that the following rule is derivable in $\mathbf{miK4}$.

$$\frac{\Gamma \vdash \bullet(A \rightarrow B)}{\Gamma \vdash \bullet A \rightarrow \bullet B} \text{ (K)}$$

Similarly, we define $\mathbf{LA}\mu$ as the formal system obtained from $\mathbf{miK4}$ by adding the following four additional rules.

$$\frac{\Gamma \vdash A[\mu X. A/X]}{\Gamma \vdash \mu X. A} \text{ (fold)} \qquad \frac{\Gamma \vdash \mu X. A}{\Gamma \vdash A[\mu X. A/X]} \text{ (unfold)} \qquad \frac{\Gamma \vdash \bullet A \rightarrow \bullet B}{\Gamma \vdash \bullet(A \rightarrow B)} \text{ (L)} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash \bullet A} \text{ (approx)}$$

We occasionally use $\vdash A$ as a synonym to $\{\} \vdash A$. Note that the (4)-rule is redundant to $\mathbf{LA}\mu$ since it has (approx). When $\mathbf{miK4}$ is considered, we usually suppose that formulae, i.e., type expressions, are finite, that is, do not include any occurrence of μ . It will be shown, later in Theorem 10.14, that the modal logic $\mathbf{LA}\mu$ corresponds to the typing system $\lambda \mathbf{A}$. In $\mathbf{LA}\mu$, (fold), (unfold), (K), (L) and (approx) substitute for the (\preceq) -rule of $\lambda \mathbf{A}$. The formal system $\mathbf{LA}\mu$ does not have a rule directly corresponding to $(\cong\text{-uniq})$ of $\lambda \mathbf{A}$. This is because the logical equivalence between formulae is a weaker notion than the one as types, i.e., sets of realizers, and is derivable from other inference rules. The first thing we should confirm is the following proposition.

Proposition 10.2. *If $\{A_1, \dots, A_n\} \vdash B$ is derivable in $\mathbf{LA}\mu$, then $\{x_1 : A_1, \dots, x_n : A_n\} \vdash M : B$ is derivable in $\lambda \mathbf{A}$ for some λ -term M and distinct individual variables x_1, \dots, x_n such that $FV(M) \subseteq \{x_1, \dots, x_n\}$.*

To show the opposite, we will employ the completeness of $\mathbf{LA}\mu$ with respect to a certain interpretation of formulae. As a preparation for that, we first show that in $\mathbf{LA}\mu$,

- Proposition 10.3.** *The following inference rule is derivable in $\mathbf{LA}\mu$.*

60

Proposition 10.6. 1. Suppose that $\Gamma' \vdash_{\mathbf{LA}\mu} \Gamma$. If $\Gamma \vdash A$ is derivable in $\mathbf{LA}\mu$, then so is $\Gamma' \vdash A$.

2. If $\{A_1, A_2, \dots, A_n\} \vdash B$ is derivable in $\mathbf{LA}\mu$, then so is $\{A_1[C/X], A_2[C/X], \dots, A_n[C/X]\} \vdash B[C/X]$ for every C and X .

Proof. Straightforward. □

Proposition 10.7. $\{A^t\} \vdash A$ is derivable in $\mathbf{LA}\mu$ for every A .

Proof. By induction on $h(A)$, and by cases on the form of A .

Case: $A = X$ for some X . Trivial since $A^t = A$ by Definition 2.2.

Case: $A = \bullet B$ for some B . In this case, $A^t = \bullet B^t$. By induction hypothesis, $\{B^t\} \vdash B$ is derivable; hence, so is $\{\bullet B^t\} \vdash \bullet B$ by (nec).

Case: $A = B \rightarrow C$ for some B and C . In this case, $A^t = C^t$. By induction hypothesis and Proposition 10.6.1, $\{B, C^t\} \vdash C$ is derivable, from which we get $\{C^t\} \vdash B \rightarrow C$ by (\rightarrow I).

Case: $A = \mu X.B$ for some X and B . In this case, $A^t = \mu X.B^t$, and by induction hypothesis,

$$\{B^t\} \vdash B \text{ is derivable.} \quad (93)$$

If $X \notin FTV(B^t)$, then $\{A^t\} \vdash B^t$ is derivable from (assump) by (unfold) since $B^t[A^t/X] = B^t$; and hence, so is $\{A^t\} \vdash B$ by (93) and Proposition 10.6.1. In this case, we can derive $\{A^t\} \vdash B[A/X]$ by Proposition 10.6.2, from which we get $\{A^t\} \vdash A$ by (fold). On the other hand, in case of $X \in FTV(B^t)$, let $B^t = \bullet^{m_0} \mu Y_1. \bullet^{m_1} \mu Y_2. \bullet^{m_2} \dots \mu Y_n. \bullet^{m_n} X$, where $X \notin \{Y_1, Y_2, \dots, Y_n\}$, and let $k = m_0 + m_1 + m_2 + \dots + m_n$. Note that A is a \top -variant and $k > 0$ in this case, since B^t is also proper in X by Proposition 2.9.2. Hence, $\{\bullet^k X\} \vdash B^t$ is derivable from $\{X\} \vdash X$ by repeatedly applying (nec) or (fold); and therefore, so is $\{\bullet X\} \vdash B^t$ by Proposition 10.6.1 since $\{\bullet X\} \vdash \bullet^k X$ is derivable from $\{\bullet X\} \vdash \bullet X$ by applying (approx) $k - 1$ times. Hence, by (93),

$$\{\bullet X\} \vdash B \text{ is also derivable.}$$

Therefore, by Proposition 10.6.2, so is $\{\bullet A\} \vdash B[A/X]$, from which we get $\{\bullet A\} \vdash A$ by (fold). Then, $\vdash \bullet A \rightarrow A$ is derivable by (\rightarrow I); and hence, so is $\vdash A$ by Proposition 10.3. Finally, we get $\{A^t\} \vdash A$ by Proposition 10.6.1. □

Proposition 10.8. If A is a \top -variant, then $\vdash A$ is derivable in $\mathbf{LA}\mu$.

Proof. Suppose that A is a \top -variant, i.e., $A^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \bullet^{m_2} \dots \mu X_n. \bullet^{m_n} X_i$ for some $n, m_0, m_1, m_2, \dots, m_n, X_1, X_2, \dots, X_n$ and i such that $1 \leq i \leq n, X_i \notin \{X_{i+1}, X_{i+2}, \dots, X_n\}$ and $m_i + m_{i+1} + m_{i+2} + \dots + m_n \geq 1$. By Proposition 10.7, it suffices to show that $\vdash A^t$ is derivable. Let k be the largest integer such that $m_k > 0$, and let C and D as $C = \mu X_{k+1}. \mu X_{k+2}. \dots \mu X_n. X_i$ and $D = \bullet^{m_i} \mu X_{i+1}. \bullet^{m_{i+1}} \mu X_{i+2}. \dots \bullet^{m_{k-1}} \mu X_k. \bullet^{m_k} C$. Note that $A^t = \bullet^{m_0} \mu X_1. \bullet^{m_1} \mu X_2. \dots \bullet^{m_{i-1}} \mu X_i. D$. We can derive $\{X_i\} \vdash C$ from $\{X_i\} \vdash X_i$ by (fold), from which $\{\bullet X_i\} \vdash \bullet C$ follows by (nec). Hence, continuing the derivation by repeatedly applying (approx) or (fold), we get $\{\bullet X_i\} \vdash D$. Therefore, by Proposition 10.6.2, $\{\bullet \mu X_i. D\} \vdash D[\mu X_i. D/X_i]$ is derivable, and from which $\{\bullet \mu X_i. D\} \vdash \mu X_i. D$ follows by (fold). Hence, $\vdash \mu X_i. D$ is derivable by Proposition 10.3, and therefore, so is $\vdash A^t$ by applying (nec) and (fold). □

Proposition 10.9. Let $[\vec{B}/\vec{X}]$ and $[\vec{C}/\vec{X}]$ be abbreviations for $[B_1/X_1, B_2/X_2, \dots, B_n/X_n]$ and $[C_1/X_1, C_2/X_2, \dots, C_n/X_n]$, respectively. Suppose that for any $i \in \{1, 2, \dots, n\}$, either

- (a) $\{B_i \rightarrow C_i, C_i \rightarrow B_i\} \subseteq \Gamma$, or
- (b) $\{\bullet(B_i \rightarrow C_i), \bullet(C_i \rightarrow B_i)\} \subseteq \Gamma$ and A is proper in X_i .

Then, $\Gamma \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$.

Proof. By induction on $h(A)$, and by cases on the form of A . If A is a \top -variant, then so are both $A[\vec{B}/\vec{X}]$ and $A[\vec{C}/\vec{X}]$ by Proposition 2.10.1; and hence, it is trivial by Proposition 10.8. Therefore, we only consider the case that A is not.

Case: $A = Y$ for some Y . Trivial if $Y \notin \{X_1, X_2, \dots, X_n\}$. If $Y = X_i$ for some i , then $A[\vec{B}/\vec{X}] = B_i$ and $A[\vec{C}/\vec{X}] = C_i$; and hence, $\Gamma \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$ from (a) since A is not proper in X_i .

Case: $A = \bullet D$ for some D . Let $\Gamma' = \{B_1 \rightarrow C_1, C_1 \rightarrow B_1, B_2 \rightarrow C_2, C_2 \rightarrow B_2, \dots, B_n \rightarrow C_n, C_n \rightarrow B_n\}$. By induction hypothesis, $\Gamma \cup \Gamma' \vdash_{\mathbf{LA}\mu} D[\vec{B}/\vec{X}] \leftrightarrow D[\vec{C}/\vec{X}]$; and hence, both $\bullet\Gamma \cup \bullet\Gamma' \vdash \bullet(D[\vec{B}/\vec{X}] \rightarrow D[\vec{C}/\vec{X}])$ and $\bullet\Gamma \cup \bullet\Gamma' \vdash \bullet(D[\vec{C}/\vec{X}] \rightarrow D[\vec{B}/\vec{X}])$ are derivable by (nec); and so are $\bullet\Gamma \cup \bullet\Gamma' \vdash A[\vec{B}/\vec{X}] \rightarrow A[\vec{C}/\vec{X}]$ and $\bullet\Gamma \cup \bullet\Gamma' \vdash A[\vec{C}/\vec{X}] \rightarrow A[\vec{B}/\vec{X}]$ by (K). Therefore, since $\Gamma \vdash_{\mathbf{LA}\mu} \bullet\Gamma \cup \bullet\Gamma'$ by (approx), $\Gamma \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$ by Proposition 10.6.1.

Case: $A = D \rightarrow E$ for some D and E . Since A is not a \top -variant, A is proper in X_i if and only if so are D and E . Therefore, $\Gamma \vdash_{\mathbf{LA}\mu} D[\vec{B}/\vec{X}] \leftrightarrow D[\vec{C}/\vec{X}]$ and $\Gamma \vdash_{\mathbf{LA}\mu} E[\vec{B}/\vec{X}] \leftrightarrow E[\vec{C}/\vec{X}]$ by induction hypothesis; and hence, $\Gamma \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$.

Case: $A = \mu Y.D$ for some Y and D . We can assume that $Y \notin FTV(\Gamma) \cup \{\vec{X}\}$ without loss of generality.

$$\begin{aligned} A[\vec{B}/\vec{X}] &= \mu Y.D[\vec{B}/\vec{X}] \\ &\leftrightarrow_{\mathbf{LA}\mu} D[\vec{B}/\vec{X}][A[\vec{B}/\vec{X}]/Y] && \text{(by (fold) and (unfold))} \\ &= D[\vec{B}/\vec{X}, A[\vec{B}/\vec{X}]/Y] && \text{(since } Y \notin FTV(\Gamma) \cup \{\vec{X}\} \text{)} \end{aligned}$$

Similarly, we get $A[\vec{C}/\vec{X}] \leftrightarrow_{\mathbf{LA}\mu} D[\vec{C}/\vec{X}, A[\vec{C}/\vec{X}]/Y]$. On the other hand, by Definition 2.4, A is proper in X_i if and only if so is D , since A is not a \top -variant. Hence, by induction hypothesis,

$$\Gamma \cup \{\bullet(A[\vec{B}/\vec{X}] \rightarrow A[\vec{C}/\vec{X}]), \bullet(A[\vec{C}/\vec{X}] \rightarrow A[\vec{B}/\vec{X}])\} \vdash_{\mathbf{LA}\mu} D[\vec{B}/\vec{X}, A[\vec{B}/\vec{X}]/Y] \leftrightarrow D[\vec{C}/\vec{X}, A[\vec{C}/\vec{X}]/Y].$$

Therefore, $\Gamma \cup \{\bullet(A[\vec{B}/\vec{X}] \rightarrow A[\vec{C}/\vec{X}]), \bullet(A[\vec{C}/\vec{X}] \rightarrow A[\vec{B}/\vec{X}])\} \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$; and hence, $\Gamma \vdash_{\mathbf{LA}\mu} A[\vec{B}/\vec{X}] \leftrightarrow A[\vec{C}/\vec{X}]$ by Proposition 10.5. \square

Lemma 10.10. *If $A \cong B$, then $A \leftrightarrow_{\mathbf{LA}\mu} B$.*

Proof. By induction on the derivation of $A \cong B$, and by cases on the rule applied last. Most cases are straightforward. Use Proposition 10.8 for the case of $(\cong \rightarrow \top)$. The only interesting case is $(\cong\text{-uniq})$. In this case, $B = \mu X.C$ for some X and C such that $A \cong C[A/X]$ and C is proper in X . By Proposition 10.9,

$$\{\bullet(A \rightarrow B), \bullet(B \rightarrow A)\} \vdash_{\mathbf{LA}\mu} C[A/X] \leftrightarrow C[B/X].$$

Since $A \leftrightarrow_{\mathbf{LA}\mu} C[A/X]$ by induction hypothesis, and since $B \leftrightarrow_{\mathbf{LA}\mu} C[B/X]$ by (fold) and (unfold), we get

$$\{\bullet(A \rightarrow B), \bullet(B \rightarrow A)\} \vdash_{\mathbf{LA}\mu} A \leftrightarrow B.$$

Therefore, $A \leftrightarrow_{\mathbf{LA}\mu} B$ by Proposition 10.5. \square

10.1. Kripke semantics of $\mathbf{LA}\mu$

Now we turn to the semantics of $\mathbf{LA}\mu$, and proceed to show the completeness.

Definition 10.11 (iGL-frames and \mathbf{LA} -frames). An *intuitionistic well-founded frame* is a triple $\langle \mathcal{W}, \triangleright, R \rangle$, which consists of a non-empty set \mathcal{W} of possible worlds and two accessibility relations \triangleright and R on \mathcal{W} such that

1. \triangleright is a (conversely) well-founded binary relation on \mathcal{W} ,
2. R is a transitive and reflexive binary relation on \mathcal{W} , and

3. $p R q \triangleright r$ implies $p \triangleright r$.

An **iGL-frame** is an intuitionistic well-founded frame that satisfies the following condition.

4. \triangleright is transitive.

An **LA-frame** is an intuitionistic well-founded frame that satisfies the following two conditions.

5. $p \triangleright q$ implies $p R q$.
6. if $p \triangleright q R q'$, then there exists some $r \in \mathcal{W}$ such that
 - (a) $p R r \triangleright q'$, and
 - (b) $r \triangleright s$ implies $q' R s$ for every $s \in \mathcal{W}$.

Conditions 1 through 4 constitute the class of frames, to which the intuitionistic variant of the logic of provability is sound and complete (cf. [29, 30]). Condition 5 means that the interpretation is hereditary with respect to the well-founded relation \triangleright as well as R . Condition 6 corresponds to Condition 2 of Definition 3.6, which indicates that \triangleright is *locally* linear.

Definition 10.12 (Kripke semantics of the logics). Let $\langle \mathcal{W}, \triangleright, R \rangle$ be an intuitionistic well-founded frame. A mapping f from \mathcal{W} to $\{\mathbf{t}, \mathbf{f}\}$ is *hereditary* if and only if

$$\text{if } p R q, \text{ then } f(p) = \mathbf{t} \text{ implies } f(q) = \mathbf{t}.$$

A mapping ξ that assigns a mapping (from \mathcal{W} to $\{\mathbf{t}, \mathbf{f}\}$) to each propositional variable, i.e., type variable, is called a *valuation*. We say a valuation ξ is *hereditary* if and only if $\xi(X)$ is hereditary for every X . In this paper, only hereditary valuations will be considered. We define a hereditary mapping $\mathcal{I}_L(A)^\xi$ from \mathcal{W} to $\{\mathbf{t}, \mathbf{f}\}$ for each formula A by extending ξ as follows, where we use $\models_p^\xi A$ to denote that $\mathcal{I}_L(A)^\xi(p) = \mathbf{t}$.

$$\begin{aligned} \models_p^\xi A & \quad (A \text{ is a } \top\text{-variant}) \\ \models_p^\xi X & \text{ iff } \xi(X)(p) = \mathbf{t} \\ \models_p^\xi \bullet A & \text{ iff } \models_q^\xi A \text{ for every } q \text{ such that } p \triangleright q \quad (\bullet A \text{ is not a } \top\text{-variant}) \\ \models_p^\xi A \rightarrow B & \text{ iff } \models_q^\xi A \text{ implies } \models_q^\xi B \text{ for every } q \text{ such that } p R q \\ & \quad (A \rightarrow B \text{ is not a } \top\text{-variant}) \\ \models_p^\xi \mu X. A & \text{ iff } \models_p^\xi A[\mu X. A/X] \quad (\mu X. A \text{ is not a } \top\text{-variant}) \end{aligned}$$

The interpretation $\models_p^\xi A$ is defined by induction on $\langle p, r(A) \rangle$, where we use the ordering \sqsupset defined as follows.

$$\langle p, A \rangle \sqsupset \langle q, B \rangle \text{ iff } p R q \text{ and } r(A) > r(B), \text{ or } p \triangleright q.$$

Note that \sqsupset is also well-founded, since so is \triangleright , and since Condition 3 of Definition 10.11 is satisfied. We write $\Gamma \models_p^\xi A$ if and only if $\models_p^\xi A$ whenever $\models_p^\xi B$ for every $B \in \Gamma$. It is easy to verify that for every $p, q \in \mathcal{W}$, ξ and A ,

$$\text{if } p R q, \text{ then } \models_p^\xi A \text{ implies } \models_q^\xi A.$$

By a discussion similar to Theorem 7.3, we can also observe the soundness of $\lambda\mathbf{A}$ as a modal logic with respect to this semantics of formulae. The proof proceeds as follows. In the sequel, we write $\models_p^\xi A$ if and only if $\models_p^\xi A$ for every $p \in \mathcal{W}$.

Proposition 10.13. *Let $\langle \mathcal{W}, \triangleright, R \rangle$ be an **LA-frame**, and ξ a hereditary valuation.*

1. $\models_p^\xi A[B/X] \text{ iff } \models_p^{\xi[\mathcal{I}_L(B)^\xi/X]} A.$

2. Let ξ' be a hereditary valuation such that $\xi(X)(q) = \xi'(X)(q)$ for every X and $q \triangleleft p$. If for every X , either (a) A is proper in X , or (b) $\xi(X)(p) = \xi'(X)(p)$, then $\models_p^\xi A$ iff $\models_p^{\xi'} A$.
3. If $A \simeq B$, then $\models^\xi A$ iff $\models^\xi B$.
4. Let $\gamma = \{X_1 \preceq Y_1, X_2 \preceq Y_2, \dots, X_n \preceq Y_n\}$. If $\gamma \vdash A \preceq B$ is derivable, and $\{X_i\} \models^\xi Y_i$ for every $i \in \{1, 2, \dots, n\}$, then $\{A\} \models^\xi B$.
5. If $\{x_1 : A_1, \dots, x_n : A_n\} \vdash M : B$ is derivable in $\lambda\mathbf{A}$, then $\{A_1, \dots, A_n\} \models^\xi B$.

Proof. The proofs are quite parallel to those of Proposition 3.10.1, Lemma 4.6, Theorems 4.8, 5.4 and 7.3, respectively. The proofs of Items 1 and 2 proceed by induction on the ordering \sqsubset of $\langle p, r(A) \rangle$. Those of Items 3, 4 and 5 are by induction on the derivations. The only non-trivial difference is the case for the (shift)-rules in the proof of Item 5. In this case, the derivation ends with

$$\frac{\{x_1 : \bullet A_1, x_2 : \bullet A_2, \dots, x_n : \bullet A_n\} \vdash M : \bullet B}{\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\} \vdash M : B} \text{ (shift)}.$$

For the given $\langle \mathcal{W}, \triangleright, R \rangle, p$ and ξ , we construct another \mathbf{LA} -frame $\langle \mathcal{W}', \triangleright', R' \rangle$ and a hereditary valuation ξ' for it by extending $\langle \mathcal{W}, \triangleright, R \rangle$ and ξ as follows.

$$\begin{aligned} \mathcal{W}' &= \{ \langle a, q \rangle \mid a \in \{0, 1\} \text{ and } q \in \mathcal{W} \} \\ \langle a, q \rangle \triangleright' \langle b, r \rangle &\text{ iff } \begin{cases} a = b = 0 \text{ and } q \triangleright r, \text{ or} \\ a = 1, b = 0 \text{ and } q R r \end{cases} \\ \langle a, q \rangle R' \langle b, r \rangle &\text{ iff } a \geq b \text{ and } q R r \\ \xi'(X)(\langle a, q \rangle) &= \xi(X)(q) \end{aligned}$$

We can easily verify that $\langle \mathcal{W}', \triangleright', R' \rangle$ satisfies Conditions 1, 2, 3 and 5 of Definition 10.11, and that ξ' is hereditary. Furthermore, Condition 6 is also satisfied. In fact, suppose that $\langle a, p' \rangle \triangleright' \langle b, q \rangle R' \langle c, q' \rangle$. If $a = 0$, then it is obvious, since $a = 0$ implies $b = c = 0$, and since $\langle \mathcal{W}, \triangleright, R \rangle$ satisfies Condition 6. On the other hand, if $a = 1$, then we get $b = c = 0$ by the definition of \triangleright' ; and hence,

$$p' R q R q'.$$

Then, it suffices to let r be $\langle 1, q' \rangle$, because

$$\langle a, p' \rangle R' \langle 1, q' \rangle \triangleright' \langle c, q' \rangle,$$

and because $\langle 1, q' \rangle \triangleright' \langle d, s \rangle$ implies $d = 0$ and $q' R s$ by the definition of \triangleright' ; and hence, $\langle c, q' \rangle R' \langle d, s \rangle$.

Now, let $\mathcal{I}'_L(A)^{\xi'}$ be the interpretation of A in $\langle \mathcal{W}', \triangleright', R' \rangle$ under the valuation ξ' . Then, $\mathcal{I}'_L(A)^{\xi'}_{\langle 0, q \rangle} = \mathcal{I}_L(A)^{\xi}_q$ for every $q \in \mathcal{W}$ and A . Since $\mathcal{I}_L(A_i)^{\xi}_p = \mathbf{t}$ implies $\mathcal{I}'_L(\bullet A_i)^{\xi'}_{\langle 1, p \rangle} = \mathbf{t}$ for every $i \in \{1, 2, \dots, n\}$, we get $\mathcal{I}'_L(\bullet B)^{\xi'}_{\langle 1, p \rangle} = \mathbf{t}$ by the induction hypothesis on the derivation. Therefore, $\mathcal{I}'_L(B)^{\xi'}_{\langle 0, p \rangle} = \mathbf{t}$, that is, $\mathcal{I}_L(B)^{\xi}_p = \mathbf{t}$. \square

Theorem 10.14. *The following three conditions are equivalent.*

- (a) $\{A_1, A_2, \dots, A_n\} \vdash B$ is derivable in $\mathbf{LA}\mu$.
- (b) $\{x_1 : A_1, x_2 : A_2, \dots, x_n : A_n\} \vdash M : B$ is derivable in $\lambda\mathbf{A}$ for some x_1, x_2, \dots, x_n and M .
- (c) $\{A_1, A_2, \dots, A_n\} \models^\xi B$ for every \mathbf{LA} -frame $\langle \mathcal{W}, \triangleright, R \rangle$ and hereditary valuation ξ .

Proof. We get (a) \Rightarrow (b) and (b) \Rightarrow (c) by Propositions 10.2 and 10.13.5, respectively. Hence, it suffices to show that (c) \Rightarrow (a), which is established by Theorem 10.16 below. \square

Note that this theorem implies that the (shift)-rule of $\lambda\mathbf{A}$ is derivable in $\mathbf{LA}\mu$, that is, if $\bullet\Gamma \vdash \bullet A$ is derivable in $\mathbf{LA}\mu$, then so is $\Gamma \vdash A$ ¹⁰.

Definition 10.15. Let A and B type expressions, i.e., formulae of $\mathbf{LA}\mu$. We call A a *component* of B , and write $A \leq B$, if and only if $C[A/X] \cong B$ for some X and C such that $X \in ETV(C)$. We also define $Comp(B)$ as

$$Comp(B) = \{ A \mid A \leq B \}.$$

That is, $Comp(B)$ is the set of all type expressions that is equivalent, in the sense of \cong , to some subexpression occurring freely and *effectively* in B . For example, let $C = \mu X. Y \rightarrow \bullet(X \rightarrow Z)$. Then, $A \in Comp(Y \rightarrow \bullet C)$ if and only if

$$A \cong A' \text{ for some } A' \in \{ Y, Z, C, \bullet C, C \rightarrow Z, \bullet(C \rightarrow Z), Y \rightarrow \bullet C \}.$$

It can be shown that for every B , $Comp(B)/\cong$ is a finite set. We refer the reader for the proof to Appendix A.2. We do not define the corresponding notions using \simeq instead of \cong . If we define such a set, say $Comp^\simeq(B)$, by \simeq , then $Comp^\simeq(B)/\simeq$ is not always finite¹¹ because of the $(\simeq\text{-K/L})$ -rule. For example, consider $B = \mu X. \bullet(X \rightarrow Y)$. Then, $A \in Comp^\simeq(B)/\simeq$ iff $A \simeq A'$ for some $A' \in \{ Y, B, B \rightarrow Y, \bullet Y, \bullet B, \bullet B \rightarrow \bullet Y, \bullet\bullet Y, \bullet\bullet B, \bullet\bullet B \rightarrow \bullet\bullet Y, \bullet\bullet\bullet Y, \bullet\bullet\bullet B, \bullet\bullet\bullet B \rightarrow \bullet\bullet\bullet Y, \dots \}$, since

$$\begin{aligned} B &\simeq \bullet(B \rightarrow Y) \\ &\simeq \bullet B \rightarrow \bullet Y \\ &\simeq \bullet\bullet(B \rightarrow Y) \rightarrow \bullet Y \\ &\simeq (\bullet\bullet B \rightarrow \bullet\bullet Y) \rightarrow \bullet Y \\ &\simeq (\bullet\bullet\bullet(B \rightarrow Y) \rightarrow \bullet\bullet Y) \rightarrow \bullet Y \\ &\simeq ((\bullet\bullet\bullet B \rightarrow \bullet\bullet\bullet Y) \rightarrow \bullet\bullet Y) \rightarrow \bullet Y \\ &\simeq \dots \end{aligned}$$

Theorem 10.16 (Kripke completeness of $\mathbf{LA}\mu$). *If $\{ A_1, A_2, \dots, A_n \} \vdash B$ is not derivable in $\mathbf{LA}\mu$, then there exist some \mathbf{LA} -frame $\langle \mathcal{W}_0, \triangleright_0, R_0 \rangle$, hereditary valuation ξ_0 , and $p_0 \in \mathcal{W}_0$ such that $\not\models_{p_0}^{\xi_0} B$ while $\models_{p_0}^{\xi_0} A_i$ for every $i \in \{ 1, 2, \dots, n \}$.*

Proof. Suppose that $\{ A_1, \dots, A_n \} \vdash B$ is not derivable in $\mathbf{LA}\mu$. Let

$$\mathcal{F} = \{ C \mid C \in Comp(B) \text{ or } C \in Comp(A_i) \text{ for some } i \},$$

and define \mathcal{W}_0 and p_0 as follows.

$$\begin{aligned} \mathcal{W}_0 &= \{ p \subseteq \mathcal{F} \mid C \in p \text{ whenever } C \in \mathcal{F} \text{ and } p \vdash C \text{ is derivable}^{12} \} \\ p_0 &= \{ C \in \mathcal{F} \mid \{ A_1, \dots, A_n \} \vdash C \text{ is derivable} \} \end{aligned}$$

Note that \mathcal{F}/\cong is a finite set, since so are $Comp(B)/\cong$ and $Comp(A_i)/\cong$ for every i ; and hence, \mathcal{W}_0 is also a finite set because $C \cong D$ implies $C \leftrightarrow_{\mathbf{LA}\mu} D$ by Lemma 10.10. Then, for each $p \in \mathcal{W}_0$, define \tilde{p} as

$$\tilde{p} = \{ C \in \mathcal{F} \mid p \vdash \bullet C \text{ is derivable} \}.$$

Observe that

$$p \vdash_{\mathbf{LA}\mu} \bullet \tilde{p} \tag{94}$$

¹⁰This is also the case for the formal systems **miGLC** and **LA**, which will be defined later.

¹¹However, $Comp(B)/\simeq$ is still finite, since $A_1 \cong A_2$ implies $A_1 \simeq A_2$.

¹²More precisely, $\Gamma \vdash C$ derivable for some finite subset Γ of p .

by the definition, and that $\tilde{p} \in \mathcal{W}_0$. For, if $\tilde{p} \vdash C$ is derivable for some $C \in \mathcal{F}$, then so is $\bullet\tilde{p} \vdash \bullet C$ by (nec); and therefore, $p \vdash \bullet C$ is also derivable, i.e., $C \in \tilde{p}$, by Proposition 10.6.1 and (94). Note also that

$$p \subseteq \tilde{p}$$

since $\mathbf{LA}\mu$ has the (approx)-rule. We then define accessibility relations \triangleright_0 and R_0 as follows.

$$\begin{aligned} p \triangleright_0 q & \text{ iff } \tilde{p} \subseteq q \text{ and } \tilde{q} \not\subseteq q \\ p R_0 q & \text{ iff } p = q \text{ or, } p \subsetneq q \text{ and } \tilde{q} \not\subseteq q \end{aligned}$$

Since \mathcal{W}_0 is finite, and since \triangleright_0 is transitive and irreflexive, \triangleright_0 is (conversely) well-founded. We can easily verify that \triangleright_0 and R_0 satisfy Conditions 1, 2, 3 and 5 of Definition 10.11. Furthermore, Condition 6 is also satisfied. For, suppose that $p \triangleright_0 q R_0 q'$. Then, let $r = \{ C \in \mathcal{F} \mid p \cup \bullet q' \vdash C \text{ is derivable} \}$. We show that (a) and (b) of Condition 6 of Definition 10.11 hold. First, obviously $p \subseteq r$ by the definition. Furthermore, it can be shown that $\tilde{r} \not\subseteq r$ by contradiction. Suppose that $\tilde{r} \subseteq r$. Since $\tilde{q}' \not\subseteq q'$ from $p \triangleright_0 q R_0 q'$, there exists some $D \in \tilde{q}'$ such that $D \notin q'$. Hence, we can derive the following.

$$\begin{aligned} q' \vdash \bullet D & \quad (\text{since } D \in \tilde{q}') \\ \bullet q' \vdash \bullet\bullet D & \quad (\text{by (nec)}) \\ r \vdash \bullet\bullet D & \quad (\text{by the definition of } r) \\ \tilde{r} \vdash \bullet D & \quad (\text{by the definition of } \tilde{r}) \\ r \vdash \bullet D & \quad (\text{since } \tilde{r} \subseteq r \text{ by assumption}) \\ \tilde{r} \vdash D & \quad (\text{by the definition of } \tilde{r}) \\ r \vdash D & \quad (\text{since } \tilde{r} \subseteq r \text{ by assumption}) \\ p \cup \bullet q' \vdash D & \quad (\text{by the definition of } r) \\ q' \cup \bullet q' \vdash D & \quad (\text{since } p \subseteq q' \text{ from } p \triangleright_0 q R_0 q') \\ q' \vdash D & \quad (\text{since } q' \vdash \bullet q' \text{ by (approx)}) \end{aligned}$$

This contradicts $D \notin q'$; and hence, $\tilde{r} \not\subseteq r$, and $p R_0 r$. Second, we show that $r \triangleright_0 q'$. To this end, suppose that $C \in \tilde{r}$. Then, the following judgments are derivable.

$$\begin{aligned} r \vdash \bullet C & \quad (\text{since } C \in \tilde{r}) \\ p \cup \bullet q' \vdash \bullet C & \quad (\text{by the definition of } r) \\ p \vdash \bullet q' \rightarrow \bullet C & \quad (\text{by } (\rightarrow I)^{13}) \\ p \vdash \bullet(q' \rightarrow C) & \quad (\text{by } (\mathbf{L})) \\ \tilde{p} \vdash q' \rightarrow C & \quad (\text{by the definition of } \tilde{p}) \\ \tilde{p} \cup q' \vdash C & \quad (\text{by } (\rightarrow E)) \\ q' \vdash C & \quad (\text{since } \tilde{p} \subseteq q' \text{ from } p \triangleright_0 q R_0 q') \end{aligned}$$

Hence, $C \in q'$; that is, $\tilde{r} \subseteq q'$; and therefore, $r \triangleright_0 q'$, since $\tilde{q}' \not\subseteq q'$ from $p \triangleright_0 q R_0 q'$, Condition (a) is thus established. Next, we show that (b) also holds. Suppose that $r \triangleright_0 s$ and $C \in q'$. It suffices to show that $C \in s$. We can derive

$$\begin{aligned} q' \vdash C & \quad (\text{since } C \in q') \\ \bullet q' \vdash \bullet C & \quad (\text{by (nec)}) \\ r \vdash \bullet C & \quad (\text{by the definition of } r) \\ \tilde{r} \vdash C & \quad (\text{by the definition of } \tilde{r}) \\ s \vdash C & \quad (\text{since } \tilde{r} \subseteq s \text{ from } r \triangleright_0 s). \end{aligned}$$

¹³More precisely, $\Gamma \vdash \bullet A_1 \rightarrow \bullet A_2 \rightarrow \dots \rightarrow \bullet A_n \rightarrow \bullet C$ is derivable for some finite subset Γ of p and some finite subset $\{A_1, A_2, \dots, A_n\}$ of q .

That is, $C \in s$. Thus, Condition (b) is also established. We finally define the valuation ξ_0 as follows.

$$\xi_0(X)_p = \begin{cases} \mathbf{t} & (X \in p) \\ \mathbf{f} & (X \notin p) \end{cases}$$

Obviously, ξ_0 is hereditary by the definition of R_0 . To finish the proof of the completeness theorem, it suffices to prove the following lemma, because $B \notin p_0$ while $A_i \in p_0$ for every i . \square

Lemma 10.17. *Consider the frame $\langle \mathcal{W}_0, \triangleright_0, R_0 \rangle$ and \mathcal{F} defined in the proof of Theorem 10.16. Let $C \in \mathcal{F}$ and $p \in \mathcal{W}_0$. Then, $C \in p$ if and only if $\models_p^{\xi_0} C$.*

Proof. The proof proceeds by induction on the ordering \sqsupset of $\langle p, r(C) \rangle$, and by cases on the form of C . If C is a \top -variant, then $C \in p$ by the definition of \mathcal{W}_0 and Proposition 10.8, and $\models_p^{\xi_0} C$ for every p by Definition 10.12. Hence, we only consider the case that C is not.

Case: $C = X$ for some X . Trivial from the definition of $\xi_0(X)$.

Case: $C = \bullet D$ for some D . For the “only if” part, suppose that $\bullet D \in p$. If $p \triangleright_0 q$, then since $\tilde{p} \subseteq q$ and since $p \vdash \bullet D$ is derivable, we get $D \in q$. Hence, $\models_q^{\xi_0} D$ by induction hypothesis. Thus, we get $\models_p^{\xi_0} \bullet D$ from $\bullet D \in p$. For the “if” part, suppose that $\models_p^{\xi_0} \bullet D$, i.e.,

$$\models_q^{\xi_0} D \text{ for any } q \text{ such that } p \triangleright_0 q. \quad (95)$$

Let

$$q = \{ E \in \mathcal{F} \mid \tilde{p} \cup \{\bullet D\} \vdash E \text{ is derivable} \}.$$

Note that $q \in \mathcal{W}_0$ and $\tilde{p} \subseteq q$. If $p \triangleright_0 q$, then $\models_q^{\xi_0} D$ by (95); therefore, $D \in q$ by induction hypothesis. Otherwise, $\tilde{q} \subseteq q$. In this case, again $D \in q$ since $D \in \tilde{q}$. Therefore, $\tilde{p} \cup \{\bullet D\} \vdash D$ is derivable in either case; and hence, we can derive the following judgments.

$$\begin{array}{ll} \tilde{p} \vdash \bullet D \rightarrow D & (\text{by } (\rightarrow\text{I})) \\ \tilde{p} \vdash D & (\text{by Proposition 10.3, namely } (\mathbf{Y})) \\ \bullet \tilde{p} \vdash \bullet D & (\text{by (nec)}) \\ p \vdash \bullet D & (\text{by (94) and Proposition 10.6.1}) \end{array}$$

Thus, we get $\bullet D \in p$ from $\models_p^{\xi_0} \bullet D$.

Case: $C = D \rightarrow E$ for some D and E . For the “only if” part, suppose that $D \rightarrow E \in p$, $p R_0 q$ and $\models_q^{\xi_0} D$. We get $D \in q$ from $\models_q^{\xi_0} D$ by induction hypothesis. On the other hand, $D \rightarrow E \in q$, since $p \subseteq q$ from $p R_0 q$. Therefore, $E \in q$; and by induction hypothesis again, $\models_q^{\xi_0} E$. Thus, we get $\models_p^{\xi_0} D \rightarrow E$. As for the “if” part, suppose that $\models_p^{\xi_0} D \rightarrow E$, i.e.,

$$\models_q^{\xi_0} D \text{ implies } \models_q^{\xi_0} E \text{ whenever } p R_0 q. \quad (96)$$

Let

$$q = \{ F \in \mathcal{F} \mid p \cup \{D, \bullet E\} \vdash F \text{ is derivable} \}.$$

Note that $q \in \mathcal{W}_0$ and $p \subseteq q$. If $\tilde{q} \subseteq q$, then we can derive

$$\begin{array}{ll} q \vdash \bullet E & (\text{by the definition of } q) \\ \tilde{q} \vdash E & (\text{by the definition of } \tilde{q}) \\ q \vdash E & (\text{since } \tilde{q} \subseteq q \text{ by assumption}) \\ p \cup \{D, \bullet E\} \vdash E & (\text{by the definition of } q) \\ p \cup \{D\} \vdash \bullet E \rightarrow E & (\text{by } (\rightarrow\text{I})) \\ p \cup \{D\} \vdash E & (\text{by Proposition 10.3, namely } (\mathbf{Y})) \\ p \vdash D \rightarrow E & (\text{by } (\rightarrow\text{I})). \end{array}$$

That is, $D \rightarrow E \in p$ in this case. On the other hand, if $\tilde{q} \not\subseteq q$, then $p R_0 q$ holds. Hence, $\models_q^{\xi_0} D$ from $D \in q$ by induction hypothesis. Then, $\models_q^{\xi_0} E$ from (96); and therefore, by induction hypothesis again, $E \in q$, i.e., $p \cup \{D, \bullet E\} \vdash E$ is derivable, which implies that so is $p \vdash D \rightarrow E$ by (\rightarrow I) and (**Y**). Thus, $D \rightarrow E \in p$ also in this case.

Case: $C = \mu X.D$ for some X and D . For the “if” part, suppose that $\models_p^{\xi_0} \mu X.D$, i.e., $\models_p^{\xi_0} D[\mu X.D/X]$. Since $r(D[\mu X.D/X]) < r(\mu X.D)$, we get $D[\mu X.D/X] \in p$ by induction hypothesis; and therefore, $\mu X.D \in p$ since **LA** μ has (fold). For the “only if” part, suppose that $\mu X.D \in p$, i.e., also $D[\mu X.D/X] \in p$ by (unfold). Hence, $\models_p^{\xi_0} D[\mu X.D/X]$ by induction hypothesis; and therefore, $\models_p^{\xi_0} \mu X.D$ by Definition 10.12. \square

This completes the proofs of Theorems 10.16 and 10.14. Since the counter model constructed in the proof of Lemma 10.17 is based on a finite frame, the logic $\mathbf{LA}\mu$ has the finite model property, and we therefore get the following corollary.

Corollary 10.18. *The following problems are decidable.*

1. *Provability in $\mathbf{LA}\mu$.*
2. *Type inhabitation in $\lambda\mathbf{A}$.*

10.2. Relationship to intuitionistic modal logic of provability

The logic $\mathbf{LA}\mu$ allows self-referential formulae. In this subsection, we show that if $\mathbf{LA}\mu$ is restricted to finite formulae, i.e., those without any occurrence of μ , then we get an intuitionistic version of the logic of provability \mathbf{GL} .

Definition 10.19 (miGL, miGLC and LA). We define **miGL**, which only allows finite formulae, to be the formal system obtained from **miK4** (Definition 10.1) by adding the following inference rule called Löb’s Principle.

$$\frac{\Gamma \vdash \bullet(\bullet A \rightarrow A)}{\Gamma \vdash \bullet A} \quad (\mathbf{W})$$

We also define two more formal systems: **miGLC** and **LA**, which again only allow finite formulae, as **miGL** + (approx) and **miGLC** + (**L**), respectively, where (approx) and (**L**) are those of Definition 10.1.

The formal system **miGL** corresponds to the minimal and implicational fragment of **GL**¹⁴, and it can be shown that **miGL** is sound and complete with respect to the Kripke semantics over **iGL**-frames (Definitions 10.11 and 10.12). The (approx)-rule of **miGLC** and **LA** is called the “Completeness Principle” in the context of the logic of provability (cf. [31]).

Note that **(W)**, i.e., Löb’s Principle, is a derivable rule of **LA** μ . In fact, $\vdash (\bullet A \rightarrow A) \rightarrow A$ is derivable in **LA** μ by Proposition 10.3, and from which we get $\vdash \bullet(\bullet A \rightarrow A) \rightarrow \bullet A$ by applying (nec) and **(K)**. Conversely, $\vdash (\bullet A \rightarrow A) \rightarrow A$ is derivable in **miGLC** and **LA** by **(W)** and (approx) as follows¹⁵.

$$\begin{array}{c}
\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet A \rightarrow A} \text{ (assump)} \quad \frac{\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet A \rightarrow A} \text{ (assump)}}{\{\bullet A \rightarrow A\} \vdash \bullet(\bullet A \rightarrow A)} \text{ (approx)} \\
\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet A \rightarrow A} \text{ (assump)} \quad \frac{\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet(\bullet A \rightarrow A)} \text{ (approx)}}{\{\bullet A \rightarrow A\} \vdash \bullet A} \text{ (W)} \\
\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet A \rightarrow A} \text{ (assump)} \quad \frac{\frac{}{\{\bullet A \rightarrow A\} \vdash \bullet A} \text{ (W)}}{\{\bullet A \rightarrow A\} \vdash A} \text{ (}\rightarrow\text{E)} \\
\frac{}{\vdash (\bullet A \rightarrow A) \rightarrow A} \text{ (}\rightarrow\text{I)}
\end{array}$$

¹⁴The intuitionistic version of **GL** was introduced by Ursini[29], who also showed its completeness and finite model property[30]. The (4)-rule is redundant if conjunctive formulae are available.

¹⁵If the system allows conjunctive formulae, the pair of (approx) and (**W**) is equivalent to the (**Y**)-rule of Proposition 10.3, which is called the “Strong Löb’s Principle”.

Table 1: Summary of the systems

	miGL	miGLC	LA	LA μ	$\lambda\mathbf{A}$	
system	logic				type	
recursive types	no			yes		
frames	well-founded					
	n/a		$locally$ linear			
interpretation	n/a	hereditary w.r.t. \triangleright				
rules	(assump)				(var)	
	$(\rightarrow\mathbf{I})$					
	$(\rightarrow\mathbf{E})$					
	(nec)				$derivable$	
	(\mathbf{K}) $drivable$				$(\simeq\mathbf{-K/L})$	
	n/a		(\mathbf{L})			
	$(\mathbf{4})$	$(approx)$				$(\preceq\mathbf{-approx})$
	n/a					
	n/a	$derivable$				(shift)
	(\mathbf{W})			$derivable$		
	$derivable$ <i>(for logical fixed-points of finite formulae)</i>			(fold/unfold)	$(\cong\mathbf{-fix})$	
				$(\cong\mathbf{-uniq})$		
theorems	$\bullet(A \rightarrow B) \rightarrow \bullet A \rightarrow \bullet B$					
	n/a		$(\bullet A \rightarrow \bullet B) \rightarrow \bullet(A \rightarrow B)$			
	$\bullet A \rightarrow \bullet\bullet A$					
	n/a	$A \rightarrow \bullet A$				
	$\bullet(\bullet A \rightarrow A) \rightarrow \bullet A$					
	n/a	$(\bullet A \rightarrow A) \rightarrow A$				

Therefore, since the only role of (fold) and (unfold) for finite formulae in the proof of Theorem 10.16 is the derivability of $(\bullet A \rightarrow A) \rightarrow A$, i.e., Proposition 10.3, which is used in the “if” part of the proof of Lemma 10.17, we get the following.

Theorem 10.20 (Kripke completeness of LA). *The formal system LA is also Kripke complete with respect to LA-frames.*

This theorem also implies, with Theorem 10.16, that $\mathbf{LA}\mu$ is a conservative extension of \mathbf{LA} . Sambin[25] proved that in the intuitionistic version of \mathbf{GL} , for any X and A such that X only occurs in A within the scope of an occurrence of the modal operator, we can construct a formula that acts as the (logically) unique fixed-point of the propositional function that maps X to A . In case of systems involving the (approx)-rule, such as \mathbf{miGLC} and \mathbf{LA} , it is also known that the fixed-point is quite simple, and can be $A[\top/X]$ (cf. [12]). For example, in such a logic, the recursive type $\mu X. A \times \bullet X$, which represents the type of infinite streams of A , is *logically* representable by $A \times \bullet \top$ ¹⁶, since $\vdash (A \times \bullet \top) \leftrightarrow (A \times \bullet(A \times \bullet \top))$. That is, as a logic, any recursive type is replaceable by a finite formula that is logically equivalent. However, this is not the case from the type theoretical point of view, since we need a fixed-point as a set of realizers.

¹⁶ A can be also a fixed-point of $X \mapsto A \times \bullet X$.

The correspondence between $\lambda\mathbf{A}$ and the logical systems can be summarized as Table 1¹⁷. As seen in this table, the logic behind $\lambda\mathbf{A}$ can be considered the minimal and implicational fragment of the logic of provability with the “Completeness Principle” and the axiom schema $(\bullet A \rightarrow \bullet B) \rightarrow \bullet(A \rightarrow B)$.

11. Concluding remarks

In this paper, a modal typed λ -calculus $\lambda\mathbf{A}$ with recursive types has been presented, and its soundness with respect to a realizability interpretation and the convergence of well-typed λ -terms according to their types have been shown. We have also shown that the modal logic behind $\lambda\mathbf{A}$ can be regarded as an intuitionistic fragment of the logic of provability, and shown its Kripke completeness with respect to intuitionistic, (conversely) well-founded and locally linear frames. By the completeness theorem, decidability of type inhabitation in $\lambda\mathbf{A}$ has been also shown. However, the decidability questions for type checking and typability of λ -terms in $\lambda\mathbf{A}$ are still open.

The connection to the logic of provability suggests certain subsystems of $\lambda\mathbf{A}$ which includes $\bullet A \preceq \bullet\bullet A$ instead of $A \preceq \bullet A$, or does not have $\bullet A \rightarrow \bullet B \preceq \bullet(A \rightarrow B)$, namely the converse of **(K)**. For example, it seems that all the examples we have seen in Section 9 can be also captured in the subsystem without $\bullet A \rightarrow \bullet B \preceq \bullet(A \rightarrow B)$. In this sense, the author does not see any apparent significance of this rule in practice so far. However, he also thinks $\lambda\mathbf{A}$, with this rule, more preferable for a basis for logic of programming, because it has much simpler formulation. Note that we would need two additional rules, namely (nec) and (subst) without the equality $\bullet A \rightarrow \bullet B \simeq \bullet(A \rightarrow B)$.

Although $\lambda\mathbf{A}$ was presented as a typed λ -calculus, the author does not think that it is directly applicable to type systems of programming languages. Since our framework can assert the convergence of derived programs, typing general recursive programs naturally requires some (classical) arithmetic as seen in the cases of the 91-function and the sieve of Eratosthenes, which would make mechanical type checking impossible. Our goal is rather to capture a wider range of programs in the proofs-as-programs paradigm and give an axiomatic semantics to them preserving the compositionality of programs. We have seen that our approach is applicable to some interesting programs such as fixed-point combinators and objects with binary methods, which have not been captured in the conventional frameworks.

References

- [1] M. Abadi, L. Cardelli, A theory of objects, Springer-Verlag, 1996.
- [2] R. M. Amadio, L. Cardelli, Subtyping recursive types, ACM Transactions on Programming Languages and Systems 15 (4) (1993) 575–631.
- [3] H. P. Barendregt, Lambda calculi with types, in: S. Abramsky, D. M. Gabbay, T. S. E. Maibaum (Eds.), Handbook of Logic in Computer Science, Vol. 2, Oxford University Press, 1992, pp. 118–309.
- [4] G. Boolos, The logic of provability, Cambridge University Press, 1993.
- [5] K. B. Bruce, L. Cardelli, B. C. Pierce, Comparing object encodings, Information and Computation 155 (1999) 108–133.
- [6] L. Cardelli, Amber, in: G. Cousineau, P.-L. Curien, B. Robinet (Eds.), Combinators and functional programming languages, Vol. 242 of Lecture Notes in Computer Science, Springer-Verlag, 1986, pp. 21–47.
- [7] F. Cardone, M. Coppo, Type inference with recursive types: syntax and semantics, Information and Computation 92 (1) (1991) 48–80.
- [8] R. L. Constable, S. Allen, H. Bromely, W. Cleveland, et al., Implementing Mathematics with the Nuprl Proof Development System, Prentice-Hall, 1986.
- [9] R. L. Constable, N. P. Mendler, Recursive definitions in type theory, in: Logics of Programs, Vol. 193 of Lecture Notes in Computer Science, Springer-Verlag, 1985, pp. 61–78.
- [10] R. L. Constable, S. F. Smith, Partial objects in constructive type theory, in: Proceedings of the 2nd IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, 1987, pp. 183–193.
- [11] B. Courcelle, Fundamental properties of infinite trees, Theoretical Computer Science 25 (1983) 95–169.
- [12] D. de Jongh, A. Visser, Embeddings of Heyting algebras, revised version, Vol. 115 of Logic Group Preprint Series, Utrecht University, 1994.
- [13] Y. Gurevich, S. Shelah, Fixed-point extensions of first-order logic, Annals of Pure and Applied Logic 32 (3) (1986) 265–280.
- [14] S. Hayashi, H. Nakano, PX: A Computational Logic, The MIT Press, 1988.

¹⁷In this table, “ n/a ” should be read as “not always” or “not available”.

- [15] R. Hindley, The completeness theorem for typing λ -terms, *Theoretical Computer Science* 22 (1983) 1–17.
- [16] R. Hindley, Curry's type-rules are complete with respect to F-semantics too, *Theoretical Computer Science* 22 (1983) 127–133.
- [17] W. A. Howard, The formulae-as-types notion of construction, in: R. J. Hindley, J. P. Seldin (Eds.), *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, 1980, pp. 480–490.
- [18] S. Kobayashi, M. Tatsuta, Realizability interpretation of generalized inductive definitions, *Theoretical Computer Science* 131 (1) (1994) 121–138.
- [19] D. C. Kozen, Results on the propositional μ -calculus, *Theoretical Computer Science* 27 (3) (1983) 333–354.
- [20] D. B. MacQueen, G. D. Plotkin, R. Sethi, An ideal model for recursive polymorphic types, *Information and Computation* 71 (1986) 95–130.
- [21] H. Nakano, A modality for recursion, in: *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, IEEE Computer Society Press, 2000, pp. 255–266.
- [22] H. Nakano, Fixed-point logic with the approximation modality and its Kripke completeness, in: *Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software*, Springer-Verlag, 2001, pp. 165–182.
- [23] C. Paulin-Mohring, Extracting F_ω 's programs from proofs in the calculus of constructions, in: *Proceedings of the 16th ACM Symposium on Principles of Programming Languages*, 1989, pp. 89–104.
- [24] V. R. Pratt, A decidable μ -calculus (preliminary report), in: *Proceedings of the 22nd IEEE Symposium on Foundation of Computer Science*, 1981, pp. 421–427.
- [25] G. Sambin, An effective fixed-point theorem in intuitionistic diagonalizable algebras, *Studia Logica* 35 (1976) 345–361.
- [26] D. S. Scott, A type-theoretical alternative to ISWIM, CUCH, OWHY, *Theoretical Computer Science* 121 (1993) 411–440.
- [27] W. W. Tait, Intensional interpretations of functionals of finite type I, *Journal of Symbolic Logic* 32 (1967) 198–212.
- [28] M. Tatsuta, Realizability interpretation of coinductive definitions and program synthesis with streams, *Theoretical Computer Science* 122 (1994) 119–136.
- [29] A. Ursini, Intuitionistic diagonalizable algebras, *Algebra Universalis* 9 (1979) 229–237.
- [30] A. Ursini, A modal calculus analogous to K4W, based on intuitionistic propositional logic, I° , *Studia Logica* 38 (1979) 297–311.
- [31] A. Visser, On the completeness principle: a study of provability in Heytings's arithmetic and extensions, *Annals of Mathematical Logic* 22 (1982) 263–295.

A. Appendix

A.1. Proof of Proposition 4.4

We prove a more general propositions as follows. The $(\cong\text{-uniq})$ -rule is crucial to the proof.

Proposition A.1. *Let n be a non-negative integer, X_1, X_2, \dots, X_n distinct type variables, and $A, B, C_1, C_2, \dots, C_n, D_1, D_2, \dots, D_n$ type expressions. Let $[\vec{C}/\vec{X}]$ and $[\vec{D}/\vec{X}]$ be abbreviations for $[C_1/X_1, C_2/X_2, \dots, C_n/X_n]$ and $[D_1/X_1, D_2/X_2, \dots, D_n/X_n]$, respectively. If $A \sim B$ and $C_i \sim D_i$ for every i ($i = 1, 2, \dots, n$), then $A[\vec{C}/\vec{X}] \sim B[\vec{D}/\vec{X}]$.*

Proof. By induction on the derivation of $A \sim B$, and by cases on the last rule applied in the derivation. Most cases are straightforward, where in the cases of $(\cong\text{-reflex})$ and $(\cong\text{-trans})$, use Lemma A.2, which will be shown later. If the last rule is $(\cong\text{-fix})$, then $A = \mu Y. A'$ and $B = A'[\mu Y. A'/Y]$ for some Y and A' . We can assume that $Y \notin \{X_i\} \cup FTV(C_i) \cup FTV(D_i)$ for every i . Then,

$$\begin{aligned}
 \mu Y. A'[\vec{C}/\vec{X}] &\sim A'[\vec{C}/\vec{X}][\mu Y. A'[\vec{C}/\vec{X}]/Y] && (\text{by } (\cong\text{-fix})) \\
 &= A'[\vec{C}/\vec{X}, \mu Y. A'[\vec{C}/\vec{X}]/Y] && (\text{since } Y \notin FTV(\vec{C})) \\
 &\sim A'[\vec{D}/\vec{X}, \mu Y. A'[\vec{C}/\vec{X}]/Y] && (\text{by Lemma A.2}) \\
 &= A'[\vec{D}/\vec{X}][\mu Y. A'[\vec{C}/\vec{X}]/Y] && (\text{since } Y \notin FTV(\vec{D})).
 \end{aligned}$$

Since $A'[\vec{D}/\vec{X}]$ is also proper in Y by Proposition 2.13.3, we now get $\mu Y. A'[\vec{C}/\vec{X}] \sim \mu Y. A'[\vec{D}/\vec{X}]$ by $(\cong\text{-uniq})$; and therefore,

$$\begin{aligned}
 A[\vec{C}/\vec{X}] &= \mu Y. A'[\vec{C}/\vec{X}] \\
 &\sim \mu Y. A'[\vec{D}/\vec{X}] \\
 &\sim A'[\vec{D}/\vec{X}][\mu Y. A'[\vec{D}/\vec{X}]/Y] && (\text{by } (\cong\text{-fix}))
 \end{aligned}$$

$$\begin{aligned}
&= A'[\vec{D}/\vec{X}, \mu Y.A'[\vec{D}/\vec{X}]/Y] && (\text{since } Y \notin FTV(\vec{D})) \\
&= A'[\mu Y.A'/Y][\vec{D}/\vec{X}] \\
&= B[\vec{D}/\vec{X}].
\end{aligned}$$

If the last rule is (\cong -uniq), then $B = \mu Y.B'$ for some Y and B' such that $A \sim B'[A/Y]$ and B' is proper in Y . We can assume that $Y \notin \{X_i\} \cup FTV(C_i) \cup FTV(D_i)$ for every i . Then,

$$\begin{aligned}
A[\vec{C}/\vec{X}] &\sim B'[A/Y][\vec{D}/\vec{X}] && (\text{by ind. hyp.}) \\
&= B'[\vec{D}/\vec{X}, A[\vec{D}/\vec{X}]/Y] \\
&= B'[\vec{D}/\vec{X}][A[\vec{D}/\vec{X}]/Y] && (\text{since } Y \notin FTV(\vec{D})) \\
&\sim B'[\vec{D}/\vec{X}][A[\vec{C}/\vec{X}]/Y] && (\text{by using Lemma A.2 twice}).
\end{aligned}$$

Note that $B'[\vec{D}/\vec{X}]$ is also proper in Y by Proposition 2.13.3. Hence, we now get $A[\vec{C}/\vec{X}] \sim \mu Y.B'[\vec{D}/\vec{X}]$ by (\cong -uniq); and therefore, $A[\vec{C}/\vec{X}] \sim B[\vec{D}/\vec{X}]$. \square

Lemma A.2. *Let n be a non-negative integer, X_1, X_2, \dots, X_n distinct type variables, and $A, C_1, C_2, \dots, C_n, D_1, D_2, \dots, D_n$ type expressions. If $C_i \sim D_i$ for every i ($i = 1, 2, \dots, n$), then $A[C_1/X_1, C_2/X_2, \dots, C_n/X_n] \sim A[D_1/X_1, D_2/X_2, \dots, D_n/X_n]$.*

Proof. By induction on $h(A)$, and by cases on the form of A . The only interesting case is when $A = \mu Y.A'$ for some Y and A' . We can assume that $Y \notin \{X_i\} \cup FTV(C_i) \cup FTV(D_i)$ for every i .

$$\begin{aligned}
A[\vec{C}/\vec{X}] &= \mu Y.A'[\vec{C}/\vec{X}] && (\text{since } Y \notin \{\vec{X}\} \cup FTV(\vec{C})) \\
&\sim A'[\vec{C}/\vec{X}][\mu Y.A'[\vec{C}/\vec{X}]/Y] && (\text{by } (\cong\text{-fix})) \\
&= A'[\vec{C}/\vec{X}, \mu Y.A'[\vec{C}/\vec{X}]/Y] && (\text{since } Y \notin \{\vec{X}\} \cup FTV(\vec{C})) \\
&\sim A'[\vec{D}/\vec{X}, \mu Y.A'[\vec{C}/\vec{X}]/Y] && (\text{by ind. hyp.}) \\
&= A'[\vec{D}/\vec{X}][\mu Y.A'[\vec{C}/\vec{X}]/Y] && (\text{since } Y \notin FTV(\vec{D})) \\
&= A'[\vec{D}/\vec{X}][A[\vec{C}/\vec{X}]/Y] && (\text{since } Y \notin \{\vec{X}\} \cup FTV(\vec{C}))
\end{aligned}$$

Since $A'[\vec{D}/\vec{X}]$ is also proper in Y by Proposition 2.13.3, we now get $A[\vec{C}/\vec{X}] \sim \mu Y.A'[\vec{D}/\vec{X}]$ by (\cong -uniq); and therefore, $A[\vec{C}/\vec{X}] \sim A[\vec{D}/\vec{X}]$. \square

A.2. Finiteness of $Comp(A)/\cong$

In this section, we show that $Comp(A)/\cong$ and $Comp(A)/\simeq$ are finite sets for every type expression A , which is parallel to the fact that the set of all subtrees of a regular tree is finite[11, 2].

Definition A.3. Let k be a non-negative integer. We write $A \leq_k B$ if and only if $C[A/X] \cong B$ for some C and X such that $dp_{\rightarrow}(C, X) \leq k$.

Note that $A \leq B$ if and only if $A \leq_k B$ for some k , by Proposition 2.15.1.

Proposition A.4. 1. $A \leq_k A$ for every k .

2. If $A \cong A' \leq_k B' \cong B$, then $A \leq_k B$.

3. If $k \leq l$, then $A \leq_k B$ implies $A \leq_l B$.

4. If $A \leq_k B$, then $ETV(A) \subseteq ETV(B)$.

5. If $A \leq_k B$ and $B \leq_l C$, then $A \leq_{k+l} C$.

Proof. Straightforward from Definition A.3. Use Proposition 4.4 for Item 2, Propositions 2.15.1, 2.12 and 4.11.1 for Item 4, and Propositions 2.15.3 and 2.15.1 for Item 5. \square

Proposition A.5. *Suppose that $A \leq_k B$.*

1. *If $B^{c\cong} = \top$, then $A^{c\cong} = \top$.*
2. *If $B^{c\cong} = \bullet^n X$, then $A^{c\cong} = \bullet^m X$ for some $m \leq n$.*
3. *If $B^{c\cong} = \bullet^n(C \rightarrow D)$, then*
 - (a) *$A^{c\cong} = \bullet^m(C' \rightarrow D')$ for some m , C' and D' such that $m \leq n$, $C' \cong C$ and $D' \cong D$, or*
 - (b) *$0 < k$, and either $A \leq_{k-1} C$ or $A \leq_{k-1} D$.*

Proof. By Definition A.3, there exist some E and Y such that

$$B \cong E[A/Y], \text{ and} \quad (\text{A.1})$$

$$dp_{\rightarrow}(E, Y) \leq k. \quad (\text{A.2})$$

Note that E is not a \top -variant from (A.2) by Propositions 2.9.5 and 2.15.1. We can assume that E is canonical by Propositions 4.19, 4.4 and 4.10. For Item 1, if $B^{c\cong} = \top$, then $\top \cong B \cong E[A/X]$ by Proposition 4.19; and hence, $A \cong \top$ by Proposition 2.10.2 and Theorem 4.14. Thus, we get $A^{c\cong} = \top$. For Item 2, suppose that $B^{c\cong} = \bullet^n X$. By Propositions 4.24.1 and 4.24.2, we get $E = \bullet^l Y$ for some $l \leq n$ from (A.1); and therefore, $\bullet^n X \cong E[A/Y] = \bullet^l A$ by Proposition 4.19, from which we get $A \cong \bullet^{n-l} X$ by Proposition 4.29.1. Thus, we get $A^{c\cong} = \bullet^m X$ taking m as $m = n - l$ by Proposition 4.21. As for Item 3, suppose similarly that $B^{c\cong} = \bullet^n(C \rightarrow D)$, where $D \not\cong \top$. By Propositions 4.21, 4.29.1 and 4.23.1, we get either

$$E = \bullet^l Y \text{ for some } l \leq n, \text{ or} \quad (\text{A.3})$$

$$E = \bullet^n(C' \rightarrow D') \text{ for some } C' \text{ and } D' \not\cong \top \quad (\text{A.4})$$

from (A.1). If (A.3) is the case, then $\bullet^n(C \rightarrow D) \cong B \cong E[A/Y] = \bullet^l A$ from (A.1); and hence, $A \cong \bullet^{n-l}(C \rightarrow D)$ by Proposition 4.29.1. Therefore, (a) holds by Proposition 4.21 considering $n - l$ as m . On the other hand, in case of (A.4), we get $\bullet^n(C \rightarrow D) \cong B \cong E[A/Y] = \bullet^n(C'[A/Y] \rightarrow D'[A/Y])$ from (A.1); and hence,

$$C \cong C'[A/Y] \text{ and } D \cong D'[A/Y]$$

by Propositions 4.29.1 and 4.29.2. Therefore, (b) holds, since (A.2) and (A.4) imply $dp_{\rightarrow}(C', Y) \leq k - 1$ or $dp_{\rightarrow}(D', Y) \leq k - 1$ by Definition 2.14. \square

Proposition A.6. *Suppose that $A \leq B$.*

1. *If $B^{c\cong} = \top$, then $A^{c\cong} = \top$.*
2. *If $B^{c\cong} = \bullet^n X$, then $A^{c\cong} = \bullet^m X$ for some $m \leq n$.*
3. *If $B^{c\cong} = \bullet^n(C \rightarrow D)$, then*
 - (a) *$A^{c\cong} = \bullet^m(C' \rightarrow D')$ for some m , C' and D' such that $m \leq n$, $C' \cong C$ and $D' \cong D$,*
 - (b) *$A \leq C$, or*
 - (c) *$A \leq D$.*

Proof. Obvious from Proposition A.5 \square

Proposition A.7. *If $A \leq_k B[C/X]$, then either*

- (a) *there exists some A' such that $A' \leq_k B$ and $A \cong A'[C/X]$, or*
- (b) *$dp_{\rightarrow}(B, X) \leq k$ and $A \leq_{k-dp_{\rightarrow}(B, X)} C$.*

Proof. We can assume that A , B and C are canonical by Propositions A.4.2, 4.19, 4.10 and 4.4. Suppose that $A \leq_k B[C/X]$. If $X \notin ETV(B)$, then $B[C/X] \cong B$ and $A[C/X] \cong A$ by Propositions 4.16, 4.11.1 and A.4.4. Hence, (a) holds considering A as A' in such a case. Therefore, we only consider the case that $X \in ETV(B)$. The proof proceeds by induction on k , and by cases on the forms of B and C .

Case: $B = \top$. In this case, $A = \top$ by Proposition A.5 since $B[C/X] = \top$; and hence, (a) holds considering \top as A' .

Case: $B = \bullet^n Y$ for some n and Y . Note that $X = Y$ by the assumption that $X \in ETV(B)$; and hence, $dp_{\rightarrow}(B, X) = 0$ by Definition 2.14. We now consider the form of C .

Subcase: $C = \top$. In this subcase, $B[C/X] = \bullet^n \top \cong \top$. Hence, (b) holds by Propositions A.4.1 and A.4.2 since $A \cong \top$ by Proposition A.5.

Subcase: $C = \bullet^m Z$ for some m and Z . In this subcase, $B[C/X] = \bullet^{n+m} Z$. Hence, $A = \bullet^l Z$ for some l such that $l \leq n + m$ by Proposition A.5. Therefore, if $l \leq m$, then $A \leq_0 C$; and hence, (b) holds, because $A \leq_0 C$ implies $A \leq_{k-0} C$ by Proposition A.4.3. Otherwise, i.e., if $l > m$, then let $A' = \bullet^{l-m} X$. Then, $A' \leq_0 B$ and $A'[C/X] = (\bullet^{l-m} X)[\bullet^m Z/X] = \bullet^l Z = A$. Thus, we get (a), since $A' \leq_0 B$ implies $A' \leq_k B$ by Proposition A.4.3.

Subcase: $C = \bullet^m(D \rightarrow E)$ for some m , D and E such that $E \not\cong \top$. Since $B[C/X] = \bullet^{n+m}(D \rightarrow E)$, by Proposition A.5, either

$$A = \bullet^l(D' \rightarrow E'), \quad l \leq n + m, \quad D' \cong D \text{ and } E' \cong E \text{ for some } l, D' \text{ and } E', \text{ or} \quad (\text{A.5})$$

$$0 < k, \text{ and either } A \leq_{k-1} D \text{ or } A \leq_{k-1} E. \quad (\text{A.6})$$

If (A.5) is the case, we get (a) or (b) similarly to the previous subcase. On the other hand, in case of (A.6), either $A \leq_{k-1} D$ or $A \leq_{k-1} E$ implies $A \leq_{k-0} C$ by Proposition A.4.5, since $D \leq_1 C$ and $E \leq_1 C$. Therefore, (b) holds in this case.

Case: $B = \bullet^n(D \rightarrow E)$ for some n , D and E such that $E \not\cong \top$. Since $B[C/X] = \bullet^n(D[C/X] \rightarrow E[C/X])$, by Proposition A.5,

$$A = \bullet^m(D' \rightarrow E'), \quad m \leq n, \quad D' \cong D[C/X] \text{ and } E' \cong E[C/X] \text{ for some } m, D' \text{ and } E', \quad (\text{A.7})$$

$$0 < k \text{ and } A \leq_{k-1} D[C/X], \text{ or} \quad (\text{A.8})$$

$$0 < k \text{ and } A \leq_{k-1} E[C/X]. \quad (\text{A.9})$$

In case of (A.7), we get (a) by Proposition A.4.3 taking A' as $A' = \bullet^m(D \rightarrow E)$. If (A.8) is the case, by induction hypothesis, we have either

(a') there exists some A' such that $A' \leq_{k-1} D$ and $A \cong A'[C/X]$, or

(b') $dp_{\rightarrow}(D, X) \leq k - 1$ and $A \leq_{k-1-dp_{\rightarrow}(D, X)} C$.

If (a') holds, we get $A' \leq_k B$ from $A' \leq_{k-1} D$ by Proposition A.4.5, since $D \leq_1 B$. Thus, (a') yields (a). On the other hand, note that $dp_{\rightarrow}(B, X) = \min(dp_{\rightarrow}(D, X), dp_{\rightarrow}(E, X)) + 1 \leq dp_{\rightarrow}(D, X) + 1$ by Definition 2.14. Hence, if (b') holds, we get $dp_{\rightarrow}(B, X) \leq k$ from $dp_{\rightarrow}(D, X) \leq k - 1$, and also get $A \leq_{k-dp_{\rightarrow}(B, X)} C$ from $A \leq_{k-1-dp_{\rightarrow}(D, X)} C$ by Proposition A.4.3. Thus, (b') yields (b). The proof for the case (A.9) is similar. \square

Proposition A.8. *If $A \leq B[C/X]$, then either*

(a) *there exists some A' such that $A' \leq B$ and $A \cong A'[C/X]$, or*

(b) *$A \leq C$.*

Proof. Obvious from Proposition A.7 \square

Lemma A.9. *If $A \leq_k \mu X.B$, then there exists some A' such that $A' \leq_k B$ and $A \cong A'[\mu X.B/X]$.*

Proof. Suppose that $A \leq_k \mu X.B$. The proof proceeds by induction on k . If $\mu X.B \cong \top$, then it suffices to let $A' = B$. In fact, $A' \leq_k B$ by Proposition A.4.1. Furthermore, $A \leq_k \mu X.B$ implies $A \cong \top$ by Propositions A.4.2, A.5 and 4.19; and hence, we get $A \cong \mu X.B \cong A'[\mu X.B/X]$ by (\cong -fix). Therefore, we only consider the case that $\mu X.B \not\cong \top$, that is, $\mu X.B$ is not a \top -variant, below. Since $\mu X.B \cong B[\mu X.B/X]$, by Propositions A.4.2 and A.7,

- (a) there exists some A' such that $A' \leq_k B$ and $A \cong A'[\mu X.B/X]$, or
- (b) $dp_{\rightarrow}(B, X) \leq k$ and $A \leq_{k-dp_{\rightarrow}(B, X)} \mu X.B$.

The proof is completed by showing that (b) also implies (a). Suppose that (b) is the case. Since $\mu X.B$ is not a \top -variant, we get $dp_{\rightarrow}(B, X) > 0$ by Propositions 2.15.7 and 2.15.9. Hence, by induction hypothesis, $A' \leq_{k-dp_{\rightarrow}(B, X)} B$, which implies $A' \leq_k B$ by Proposition A.4.3, and $A \cong A'[\mu X.B/X]$ for some A' . \square

Proposition A.10. *$Comp(A)/\cong$ and $Comp(A)/\simeq$ are finite sets.*

Proof. It suffices to show that $Comp(A)/\cong$ is finite, since \cong implies \simeq . The proof proceeds by induction on $h(A)$, and by cases on the form of A .

Case: $A = X$ for some X . In this case, by Proposition A.6, $B \leq A$ if and only if $B \cong X$. Therefore, $Comp(A)/\cong$ is a singleton.

Case: $A = \bullet C$ for some C . In this case, since $\bullet C = (\bullet X)[C/X]$, by Proposition A.8, $B \leq A$ implies

- (a) there exists some B' such that $B' \leq \bullet X$ and $B \cong B'[C/X]$, or
- (b) $B \leq C$.

Note that (a) implies $B' \cong \bullet X$ or $B' \cong X$ by Proposition A.6. Hence, $Comp(A)/\cong$ is finite by Proposition 4.4, since so is $Comp(C)/\cong$ by induction hypothesis.

Case: $A = C \rightarrow D$ for some C and D . Similarly, by Proposition A.6, $B \leq A$ implies either (a) $B \cong A$, (b) $B \leq C$, or (c) $B \leq D$. Hence, $Comp(A)/\cong$ is finite since so are $Comp(C)/\cong$ and $Comp(D)/\cong$ by induction hypothesis.

Case: $A = \mu X.C$ for some X and C . In this case, by Lemma A.9, $B \leq A$ implies there exists some B' such that $B' \leq C$ and $B \cong B'[A/X]$. Hence, $Comp(A)/\cong$ is finite since so is $Comp(C)/\cong$ by induction hypothesis. \square